SIEMENS

SIMATIC PCS 7 OSx

Recipe Manual

Order Number: 6ES7 6550XX058BC3
Manual Assembly Number: 2811150–0001
Original Edition

⚠ DANGER

DANGER indicates an imminently hazardous situation that, if not avoided, will result in death or serious injury.

DANGER is limited to the most extreme situations.

⚠ WARNING

WARNING indicates a potentially hazardous situation that, if not avoided, could result in death or serious injury, and/or property damage.

⚠ CAUTION

CAUTION used with a safety alert symbol indicates a potentially hazardous situation that, if not avoided, could result in minor or moderate injury.

CAUTION

CAUTION used without the safety alert symbol indicates a potentially hazardous situation that, if not avoided, could result in property damage.

NOTICE

NOTICE indicates a potential situation that, if not avoided, could result in an undesirable result or state.

Copyright 2002 by Siemens Energy & Automation, Inc.
All Rights Reserved — Printed in USA

Reproduction, transmission, or use of this document or contents is not permitted without express consent of Siemens Energy & Automation, Inc. All rights, including rights created by patent grant or registration of a utility model or design, are reserved.

Since Siemens Energy & Automation, Inc., does not possess full access to data concerning all of the uses and applications of customer's products, we do not assume responsibility either for customer product design or for any infringements of patents or rights of others which may result from our assistance.

MANUAL PUBLICATION HISTORY

SIMATIC PCS 7 OSx 4.1.2 Recipe Manual
Order Manual Number: 6ES7 6550XX058BC3
Refer to this history in all correspondence and/or discussion about this manual.

Event	Date	Description
Original Issue	7/02	Original Issue (2811150–0001)

LIST OF EFFECTIVE PAGES

Pages	Description	Pages	Description
Cover/Copyright	Original		
History/Effective Pages	Original		
iii — xiii	Original		
1-1 — 1-17	Original		
2-1 — 2-5	Original		
3-1 — 3-6	Original		
4-1 — 4-11	Original		
5-1 — 5-15	Original		
6-1 — 6-6	Original		
7-1 — 7-16	Original		
A-1 — A-4	Original		
Index-1 — Index-3	Original		
Registration	Original		

Trademarks

SIMATIC®, SINEC®, and STEP® are registered trademarks, and S5™ and S7™ are trademarks, of Siemens AG.

PCS™, APT™, Series 505™, and TISOFT™ are trademarks of Siemens Energy & Automation, Inc.

Adobe® and Acrobat® are registered trademarks of Adobe Systems, Inc.

@aGlance™ and Net OLE™ are trademarks of Axeda, Inc.

Epson® is a registered trademark of Seiko Epson Kabushiki Kaisha.

Excel™ is a trademark, and Windows® and MS-DOS® are registered trademarks, of Microsoft Corporation.

HP®, DeskJet®, LaserJet®, and PaintJet® are registered trademarks of Hewlett‑Packard Company.

IBM® is a registered trademark of International Business Machines Corporation.

Intel® is a registered trademark of Intel Corporation.

Internet® is a registered trademark of Internet, Inc.

Lantronix® is a registered trademark of Lantronix.

Linux® is a registered trademark of Linus Torvalds.

Lotus® and 1‑2‑3® are registered trademarks of Lotus Development Corporation.

Network Computing Devices® is a registered trademark of Network Computing Devices, Inc.

Oracle® is a registered trademark of Oracle Corporation.

PostScript® is a registered trademark of Adobe Systems, Inc.

Red Hat® is a registered trademark of Red Hat, Inc.

TI™ is a trademark of Texas Instruments, Inc.

Tektronix® is a registered trademark of Tektronix, Inc.

UNIX® is a registered trademark of X/Open Company, Ltd.

VMS® is a registered trademark of Compaq.

X Window System™ is a trademark, and Motif® is a registered trademark, of the Open Group.

XESS® is a licensed, registered trademark, and AIS® is a registered trademark of Applied Information Systems, Inc.

Other trademarks are the acknowledged property of their respective holders.

Contents

Preface ..... **ix**

Chapter 1 Guidelines for Recipe Configuration **1-1**

1.1 Introduction **1-2**

Recipes in the Controller 1-2
OSx Recipe Feature 1-4
Subrecipes 1-6
Configuring on Two Stations 1-7

1.2 Building Recipes **1-8**

Chapter 2 Defining Area Tags **2-1**

2.1 Overview **2-2**

2.2 Choosing Area Tag Data **2-3**

2.3 Defining Tag Attribute Data **2-4**

2.4 Deleting an Area Tag **2-5**

Chapter 3 Defining Recipe Areas **3-1**

3.1 Recipe Area Configuration Overview **3-2**

Identifying Recipe Areas 3-2
Preparing to Configure Recipe Areas 3-2
Configuring on Two Stations 3-3

3.2 Configuring Recipe Areas **3-4**

Entering the Area Name 3-4
Adding a Component 3-4
Modifying Components, Downloading Tags and Attributes 3-6

Chapter 4 Using Recipe Control Mechanisms **4-1**

4.1 Recipe Control Mechanisms **4-2**

Using Subrecipes for Control 4-2
Using Recipe Control Blocks (RCBs) 4-4
RCB_WAIT 4-4
RCB_BEG and RCB_END 4-5

4.2 Using RCBs to Control Recipe Downloads **4-6**

Nesting Recipe Control Blocks 4-6
Handshake Communication 4-6
Phased Download 4-10

Chapter 5 Defining Recipes 5-1

5.1 Recipe Configuration Overview 5-2
Overview 5-2
Accessing the Recipe Configuration Menu Display 5-2

5.2 Using the Recipe Configuration Menu Display 5-4
Selecting an Option 5-4
Specifying the Recipe/Subrecipe Name 5-4
Specifying the Entry Type 5-5
Specifying the Recipe/Subrecipe Description 5-5
Adding a Recipe/Subrecipe 5-5
Deleting a Recipe/Subrecipe 5-6
Copying a Recipe/Subrecipe 5-6
Modifying a Recipe/Subrecipe 5-8

5.3 Configuring Recipes 5-10
Adding Components 5-10
Choosing the Data Type 5-12
Entering Range Limits 5-13
Entering the Component Value 5-13
Entering Units 5-13
Setting the Scale Flag 5-13
Modifying Component Values 5-14
Linking Recipes and Recipe Areas 5-15

Chapter 6 Validating Recipes 6-1

6.1 Recipe Validation Overview 6-2
Purpose 6-2
Preparing to Validate Recipes 6-2

6.2 Validating Recipe Components 6-3
Creating a Production Recipe 6-3
Evaluating Validation Results 6-6

Chapter 7 Selecting and Loading Recipes 7-1

7.1 Selecting Recipes for Download 7-2
Preparing to Download Recipes 7-2
Accessing the Recipe Selection Display 7-2
Planning a Recipe Download 7-4
Clearing the Selected Recipe 7-5
Loading a Recipe 7-6

7.2 Loading Recipes: Troubleshooting 7-8
Loading Recipes: Bad Scale Factor 7-8
Loading Recipes: No Production Recipe Display 7-10
Loading Recipes: Delay for Role Change 7-10

7.3 Manually Loading Recipes for Production 7-11
Overview of the Download Process 7-11
Modifying Production Recipe Values 7-11
Modifying Scale Values 7-12
Loading a Production Recipe 7-13
Printing a Production Recipe 7-13
Canceling a Download Request 7-13
Suspending a Recipe Download 7-13
Continuing a Recipe Download 7-13

7.4 Loading Recipes by Controller Action 7-14
Overview of the Download Process 7-14
Using the Controller to Select and Download Recipes 7-14
Using the Controller to Cancel a Download Request 7-14
Using the Controller to Download Recipes Already Selected 7-15
Using the Controller to Modify Recipe Values 7-15

7.5 Acknowledging Recipe Alarm Tags 7-16

Appendix A Recipe Area Table A-1

A.1 Area Attributes A-2

A.2 Bit Locations for Area Attributes A-3

Index Index-1

List of Figures

1-1 Sugar Cookie Recipe Example 1-2
1-2 Controller Recipe Values 1-2
1-3 Example of Sugar Cookie Recipe Process 1-3
1-4 Typical Recipe 1-4
1-5 Flow of Recipe Data 1-5
1-6 Subrecipe Contents 1-6
1-7 Recipe with Subrecipe Inserted 1-6
1-8 Building a Recipe 1-8
1-9 Recipe Area Tag 1-9
1-10 Recipe Area and Memory 1-10
1-11 Sample Process and Recipe 1-11
1-12 Recipe Areas 1-12
1-13 Linking Recipes and Recipe Areas 1-13
1-14 Recipe Configuration Editor Display 1-14
1-15 Recipe Validation Display 1-15
1-16 Recipe Selection/Production Recipe Preparation Screens 1-16
1-17 Sample Recipe and Recipe Areas 1-17

3-1 Recipe Area and Component Configuration Display 3-2
3-2 Add Recipe Component Dialog Box 3-5
3-3 Modify Recipe Component Dialog Box 3-6

4-1 Recipe-to-Subrecipe Organization Example 4-3
4-2 Recipe Using RCB_WAIT 4-4
4-3 Recipe with RCB_BEG and RCB_END 4-5
4-4 Examples of Valid and Invalid Nesting of RCBs 4-6
4-5 Handshake Tag and Communication Bits 4-7
4-6 Using Control Blocks with Duplicate Components 4-10
4-7 Setting Necessary Values in a Control Block 4-11

5-1 Recipe Configuration Menu Display 5-3
5-2 Add Recipe Dialog Box 5-5
5-3 Copy Recipe to Dialog Box 5-7
5-4 Recipe Configuration Editor Display 5-8
5-5 Add Component Values Dialog Box 5-10
5-6 Modify Component Values Dialog Box 5-14
5-7 Linking Recipes and Recipe Areas 5-15

6-1 Recipe Validation Display 6-4

7-1 Recipe Directory 7-2
7-2 Recipe Selection Display 7-3
7-3 Selecting a Recipe for Download 7-4
7-4 Downloading a Recipe 7-5
7-5 Production Recipe Preparation Display 7-7
7-6 Recipe Scale Field Contains NaN 7-8
7-7 Modify Scale Dialog Box for Recipes 7-9
7-8 Modify Production Value Dialog Box 7-11
7-9 Modify Scale Dialog Box 7-12

List of Tables

2-1 Tag Header Fields 2-3

6-1 Examples of Possible Problems 6-5

Preface

New Features of PCS 7 OSx

SIMATIC PCS 7 OSx Release 4.1.2 supports the following new features:

- **OSx Merge Utility** — This utility allows you to merge the configuration from one OSx system (or a subset of that system) into another, currently running OSx system. This means that you can do major configuration development outside of an OSx system that is running a process, and then add it in without shutting down the process.
- **Remote computer data archiving** — Data archives can be stored on a remote computer. This can be any computer system that can support an FTP server; for example, UNIX, Linux, Windows NT, Windows 2000.
- **Permanent select list** — This feature allows you to choose whether the select list for tag details, graphics, reports, and so on (accessed from the Directory button) remains on the screen until you dismiss it, or disappears when you select an entry.
- **Graphic/tag cross reference report** — A standard report cross-referencing tags in graphics, by tag and by graphic, is available.
- **Internet Protocol netmask configuration** — You are prompted to specify the netmask value or select a default at installation.
- **Save new tag install file to hard disk** — The feature allows saving a tag file to hard disk in addition to MO disk and diskette.
- **SIMATIC Rack PC 840 support** — The Rack PC 840 will now be supported as a system unit.
- **1.3 gigabyte and 640 megabyte MO disk support** — Data Archiving and Backup/Restore will support larger MO disks with the Rack PC 840 hardware platform. Earlier 230 and 540 megabyte MO disks are still supported as well.
- **Additional printer support** — New printers in the Hewlett-Packard DeskJet line are supported.

Conventions Used in the Manual Set

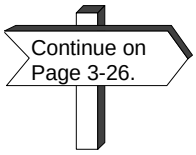
The procedures in the various manuals give you step-by-step instructions about how to carry out tasks. Typically, the last step of any procedure requires that you select the **OK** or **Save** button, or press **Enter**. To save space and avoid redundancy, this last step does not appear in the procedure. However, you need to finish each procedure with one of these actions.

OK Saves information that you have entered and closes the window.

Save Saves information that you have entered and does not close the window.

Cancel Closes the window without saving any information that you entered and terminates any action that you initiated.

Dismiss Closes the window without undoing any changes that you have entered. However, if you press **Dismiss** before you press **Enter**, the changes that you made are discarded.

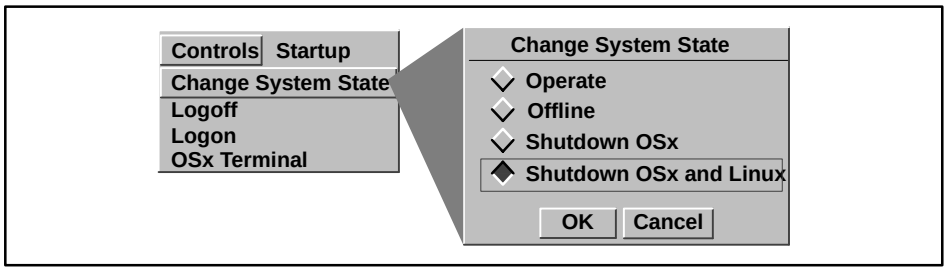


The signpost indicates that the procedure that you are currently following continues on the indicated page.

The different fonts used in the manual set have the following meanings.

- Entries that you type from the keyboard are indicated with the `courier font`.
- Items that you select on the screen, or keys that you press on the keyboard, are indicated with this **bolded font.**

Items that you select on a cascaded menu are linked in the manual text with arrows. The first term indicates where to click the main menu bar. For example, **Controls->Change System State** tells you to click **Controls** on the main menu bar, then select **Change System State** from the pull-down menu.



Purpose of This Manual

The *SIMATIC PCS 7 OSx Recipe Manual* describes how to create recipes in both SIMATIC PCS 7 OSx and the SIMATIC Application Productivity Tool — APT environments and how to use recipes for your production process.

- Chapter 1 explains recipe features and guidelines for creating recipes.
- Chapter 2 describes how to define area tags and tag attributes.
- Chapter 3 describes how to identify and configure recipe areas and how to add and modify recipe components.
- Chapter 4 describes how to use recipe control mechanisms to control recipe downloads.
- Chapter 5 describes the recipe configuration process, such as how to add, modify, copy, and delete recipes and subrecipes.
- Chapters 6 explains how to validate recipes and recipe components and how to create a production recipe.
- Chapter 7 describes how to select and download recipes to the controller.
- The Master Index contains references for all the software configuration manuals in the set.

Other Manuals

The SIMATIC PCS 7 OSx manual set consists of several manuals. If you cannot find the information that you need in the *SIMATIC PCS 7 OSx Recipe Manual*, check these other books:

- *SIMATIC PCS 7 OSx System Administration Manual* This manual describes configuring network stations, and procedures that explain how to configure printers, how to archive data, and how to back up files. Other functions normally carried out by the system administrator are also described here.
- *SIMATIC PCS 7 OSx Process Configuration Manual* This manual describes the primary tasks required to configure your OSx system for controlling your process.
- *SIMATIC PCS 7 OSx Hardware Manual* This manual describes the various hardware components of the system and how to install them.
- *SIMATIC PCS 7 OSx Graphical Editor Manual* This manual describes how to create the graphical displays that are used with OSx.
- *SIMATIC PCS 7 OSx Reports Manual* This manual describes how to create reports on your process and your OSx configuration.
- *SIMATIC PCS 7 OSx Batch Programming Manual* This manual describes more advanced configuration tasks involving the use of the Batch Control Language (BCL) and creating batch programs.
- *SIMATIC PCS 7 OSx Operator Manual* This manual is written for the configuration engineer, but it describes how to carry out the various tasks that the process operator must do when the system is in the Operate state. You can photocopy all or portions of this manual as a reference for your operators. You may prefer to rewrite sections and perhaps incorporate additional information where necessary.
- *SIMATIC PCS 7 OSx Interface to S5 Controllers Manual* This manual describes the SIMATIC PCS 7 OSx interface with SIMATIC S5 controllers.
- *SIMATIC PCS 7 OSx Interface to S7 Controllers Manual* This manual describes the OSx interface with SIMATIC S7 controllers.
- *SIMATIC PCS 7 OSx Library Manual* This manual describes the function blocks used to program the S7-400 controllers to interface with OSx.

Be sure to check the Readme File for information that did not become available until after the publication deadlines for the OSx manuals. The Readme File also points to important copyright, licensing, and warranty information. Select **Help->About OSx** from the main menu bar, and then click the **Show Readme** button at the bottom of the About OSx dialog box.

Optional SIMATIC PCS 7 OSx Features

The following manuals are available for optional SIMATIC PCS 7 OSx features.

- *SIMATIC PCS 7 OSx Remote Data Transfer Manual* This manual describes the remote data transfer feature, which allows you to transmit data collected from the process by an OSx station to an Oracle database on the remote computer for historical records and other purposes.
- *SIMATIC PCS 7 OSx X Terminal User Manual* This manual describes how to connect and operate an X terminal as an extension of an OSx station.
- *SIMATIC PCS 7 OSx @aGlance User Manual* This manual describes how to import OSx data into a Windows application, such as Excel or Lotus 1-2-3, or into another UNIX or VMS application.

If You Need Help

If you have difficulty with your system, contact the Siemens Energy & Automation, Inc., Technical Services Group in the U.S.A. at 800-333-7421. Outside the U.S.A., call 49-911-895-7000.

Chapter 1

Guidelines for Recipe Configuration

1.1 Introduction 1-2
Recipes in the Controller 1-2
OSx Recipe Feature 1-4
Subrecipes 1-6
Configuring on Two Stations 1-7

1.2 Building Recipes 1-8

1.1 Introduction

Recipes in the Controller

With controllers automating the production of specific products in your factory, you can specify values of weight, volume, mix time, cook time, temperature, and other variables by assigning those values to memory locations in the controller.

The capacity of the recipe system allows a maximum of 500 recipes and a maximum of 30,000 recipe components (or ingredients), of which 7,500 can be unique. For example, you can have 500 recipes with 60 components each, or 100 recipes with 300 components each.

For example, if you are producing sugar cookies, you might start with a basic recipe such as the one in Figure 1-1.

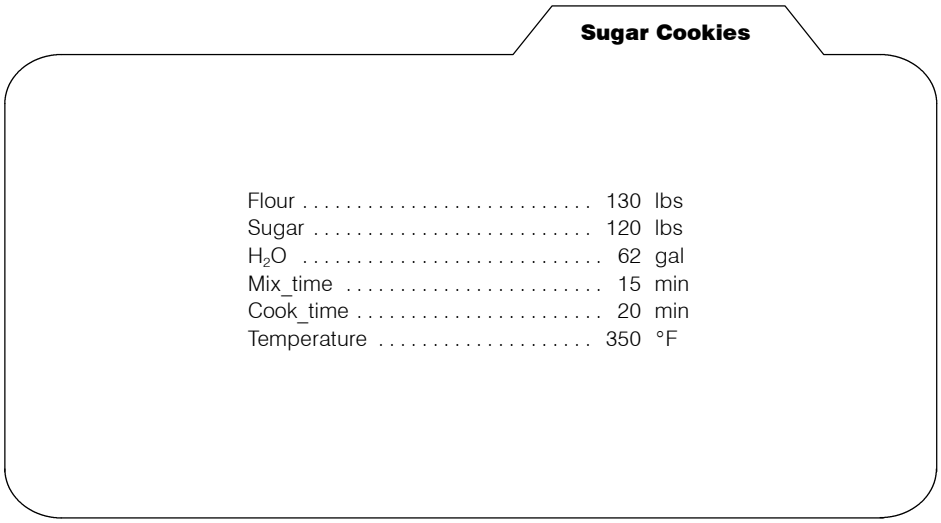


Figure 1-1 Sugar Cookie Recipe Example

In this example, you enter these ingredient values as preset values into the appropriate memory locations in the controller (Figure 1-2).

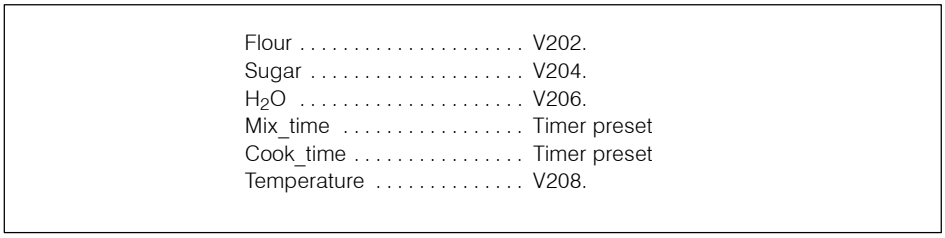


Figure 1-2 Controller Recipe Values

The controller opens and closes valves, turns pumps on or off, and activates mixers according to the controller program. Figure 1-3 represents the connection between the recipe and the equipment.

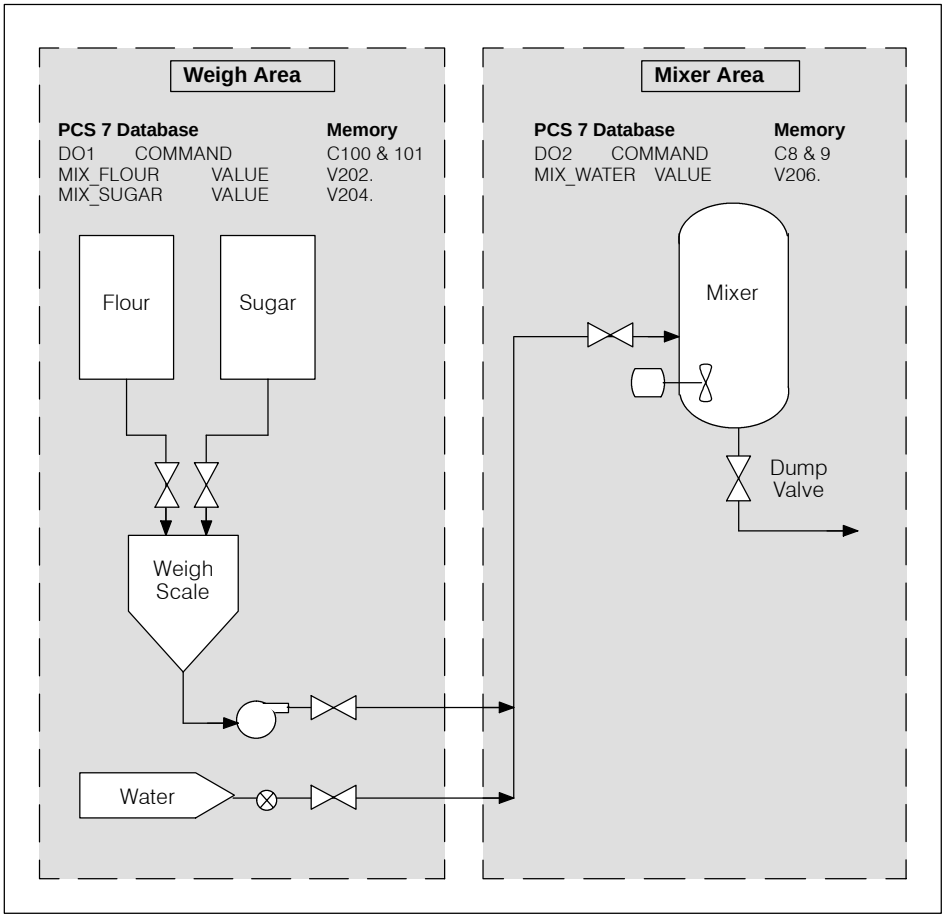


Figure 1-3 Example of Sugar Cookie Recipe Process

OSx Recipe Feature

The recipe feature allows more than the ability to store and download parameters in the process. The recipe feature allows either the operator or the control node to initiate the download of a recipe to the controller. You can use the same recipe across multiple sets of equipment. You can make modifications to the production recipe without changing the original recipe definition. In effect, the production recipe is a working copy of the original recipe.

The recipe feature allows you to configure the sets of data and the operations required to control the production of a particular type of product. These are referred to as recipes (Figure 1-4).

Recipe components define the data values that the program in the controller uses for production of a batch; for example, how much material to release from a storage bin into a mixer, how long to mix and cook, and what temperatures to use.

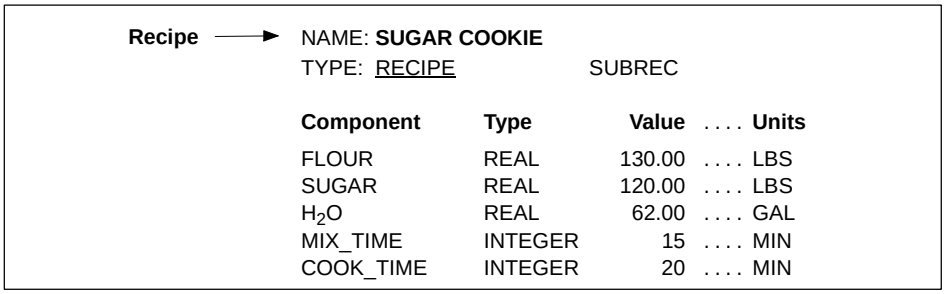


Figure 1-4 Typical Recipe

The database stores the recipe values that you configure. When you request to load a recipe, the values are downloaded to a controller that handles the production (Figure 1-5).

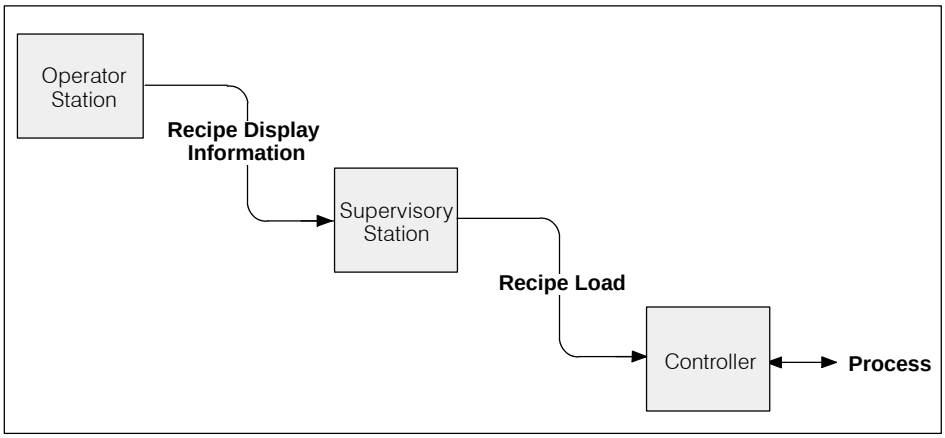


Figure 1-5 Flow of Recipe Data

The recipe management feature allows you to manipulate your production recipe in these ways:

- You can modify production recipe values.
- You can print the production recipe before downloading.
- The system logs value changes and downloads the value.

Subrecipes

If you make several products based on one or more generic recipes, you can take advantage of the subrecipe feature. The subrecipe is a configuration tool for grouping components common to more than one recipe. In Figure 1-6, flour and water are the common components used to create a subrecipe.

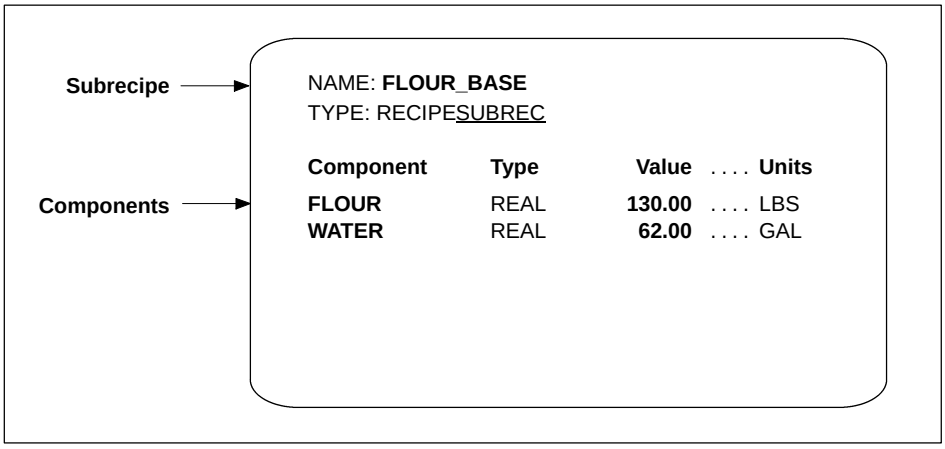


Figure 1-6 Subrecipe Contents

After configuring a subrecipe, you can insert it into any recipe (Figure 1-7).

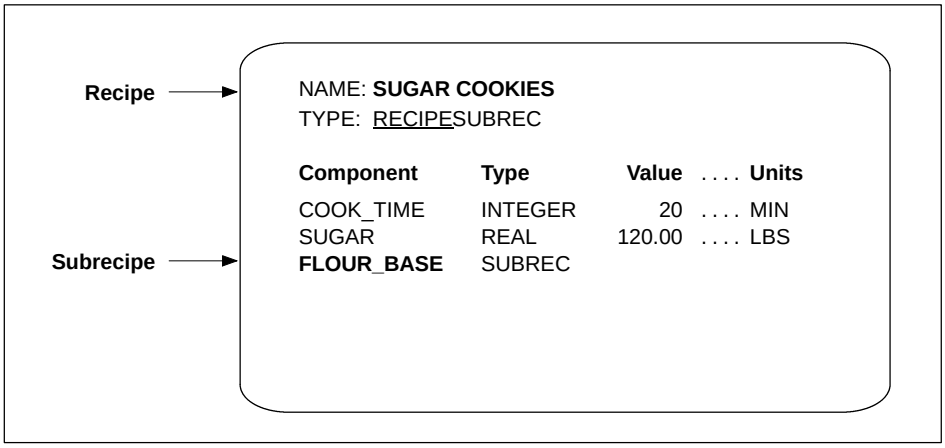


Figure 1-7 Recipe with Subrecipe Inserted

Configuring on Two Stations

It is possible for one user to start Recipe Configuration on one station and for another user to start Recipe Area and Component Configuration on another station in the system at the same time. When this happens, nothing prevents the users from working on the same recipe area at the same time.

Be aware that changes made in one configuration utility (for example, new components added to an area in Recipe Area and Component Configuration) do not show up in the other one (for example, the select list of components for an area in the Add Component Values area of Recipe Configuration) until the users exit and reenter.

1.2 Building Recipes

There are two five-step procedures that you can use to build recipes. The first procedure, described in this section, allows you to build recipes entirely within the PCS 7 environment using preconfigured I/O tags (LOOP, AI, or CALC). The second procedure allows you to build recipes in APT and then transfer the **install.tag** file to OSx (refer to the *SIMATIC APT Programming Reference Tables Manual* for more information on this procedure). Both procedures use SIMATIC PCS 7 OSx for selecting and downloading the recipes to the controller.

Figure 1-8 shows the two procedures for creating recipes in OSx and APT. Before you begin either of the procedures, set up security privileges for configuring recipes.

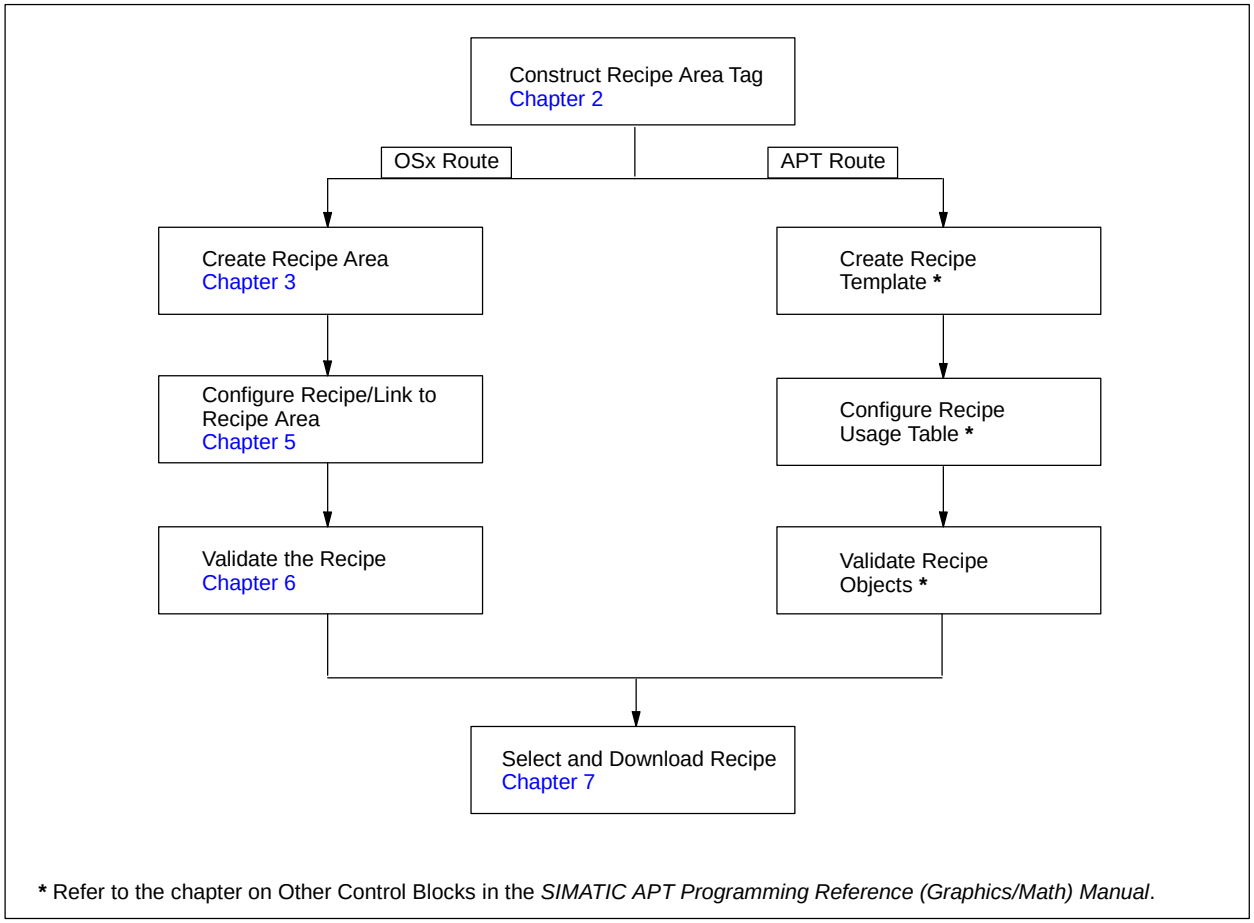
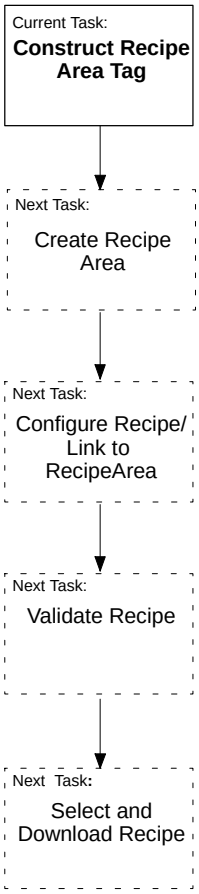


Figure 1-8 Building a Recipe



The first step in building a recipe within SIMATIC PCS 7 OSx is to construct a recipe area tag. Area tags are created by the same method that you use to create the other process I/O tags. Choose one of the following methods for configuring tags.

- APT
- Spreadsheet that accepts a comma-separated value (csv) format, such as Excel
- ASCII editor
- OSx Tag Configurator

The *SIMATIC PCS 7 OSx Process Configuration Manual* chapter on configuring tags describes these methods. For APT, refer to the *SIMATIC APT Programming Reference (Tables) Manual*.

The recipe area tag and the recipe area work together to provide a link between the OSx tag/attribute database and the components you identify for your individual recipes. The recipe area tag supports the recipe area in this function (Figure 1-9).

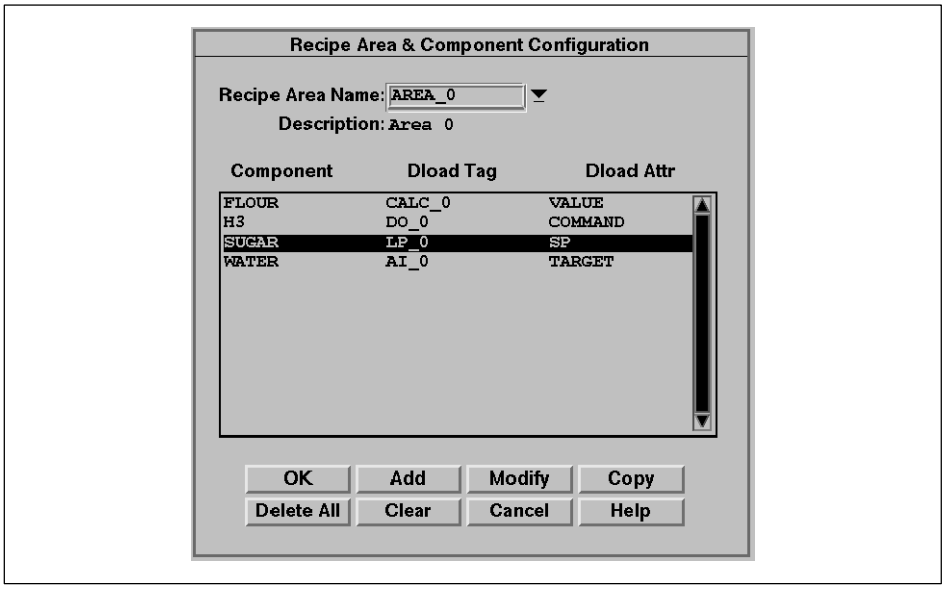
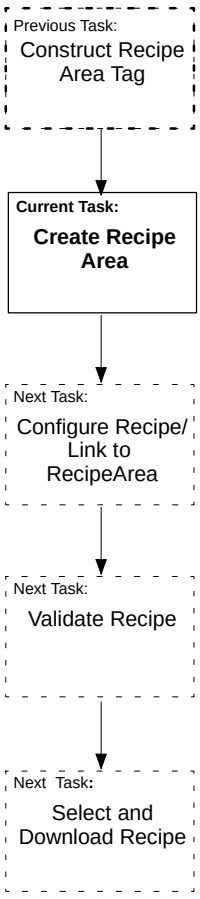


Figure 1-9 Recipe Area Tag



The second step in recipe configuration is creating a recipe area. The recipe area allows you to classify tags and attributes so that they can be addressed by the recipe management feature. The recipe area is a collection of components linked to tags/attributes that receive particular recipe values for your recipes. The components consist of ingredients, processes, processing parameters, and recipe controls. The recipe area includes all the items necessary to produce various recipes.

To construct a recipe area, identify the component, tagname, and attribute name to receive particular recipe values. The value sent to the tagname/attribute is also sent to the controller (Figure 1-10).

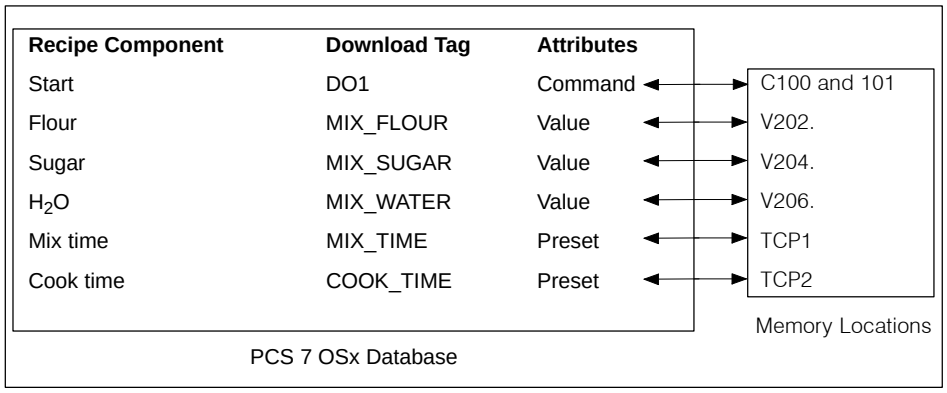


Figure 1-10 Recipe Area and Memory

As a guideline, group the tags/attributes into recipe areas that allow logical mapping of production values to tags for production of a batch. For example, Figure 1-11 outlines logical work areas that map to the two routes available for processing a batch.

As shown in the example in Figure 1-11, you can associate recipe areas with various clusters of physical equipment. This allows you to use the recipe management software to define parameters along the path for making the batch.

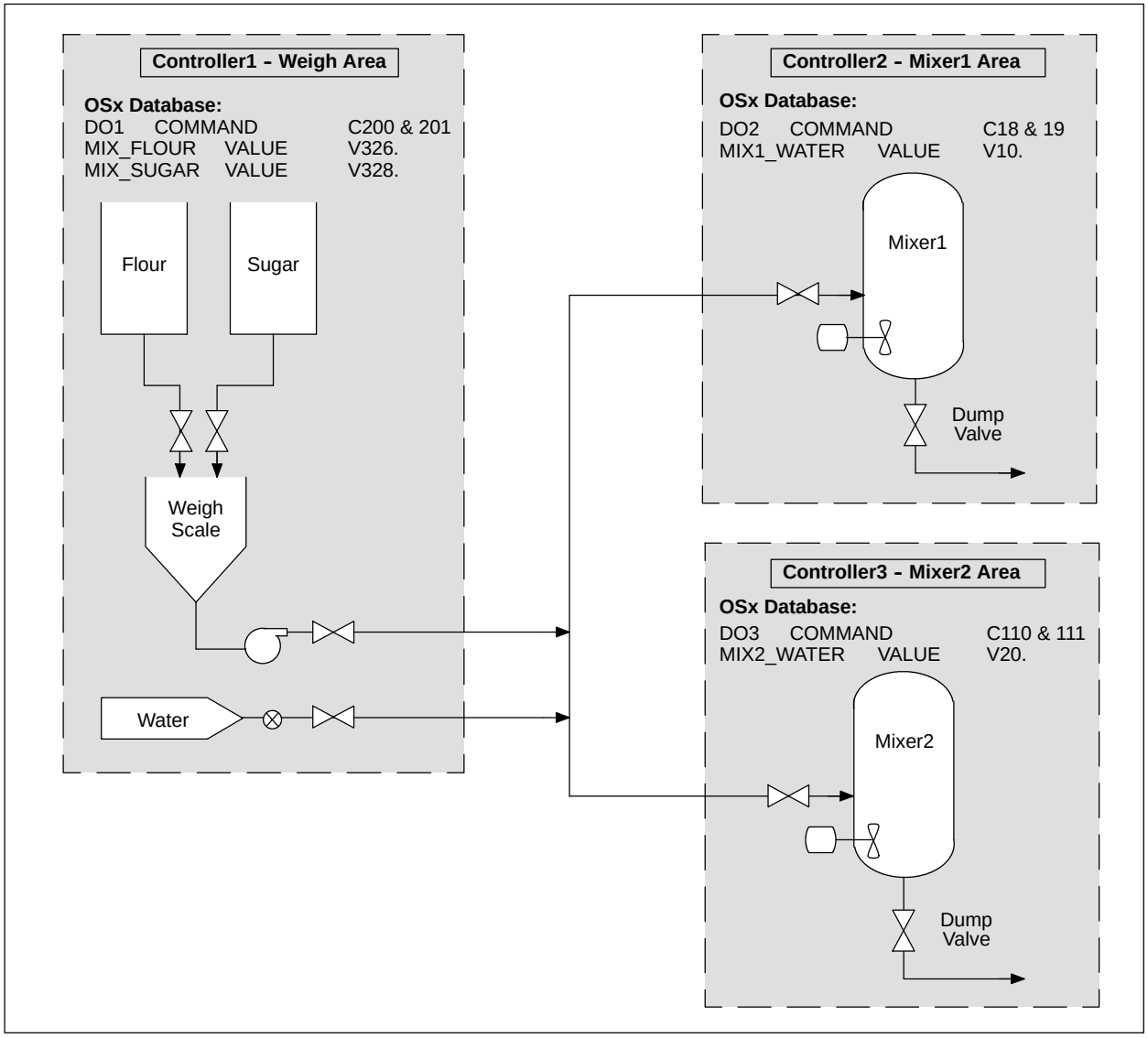


Figure 1-11 Sample Process and Recipe

Figure 1-12 shows recipe area mapping. Notice that some of the same tags appear in both areas.

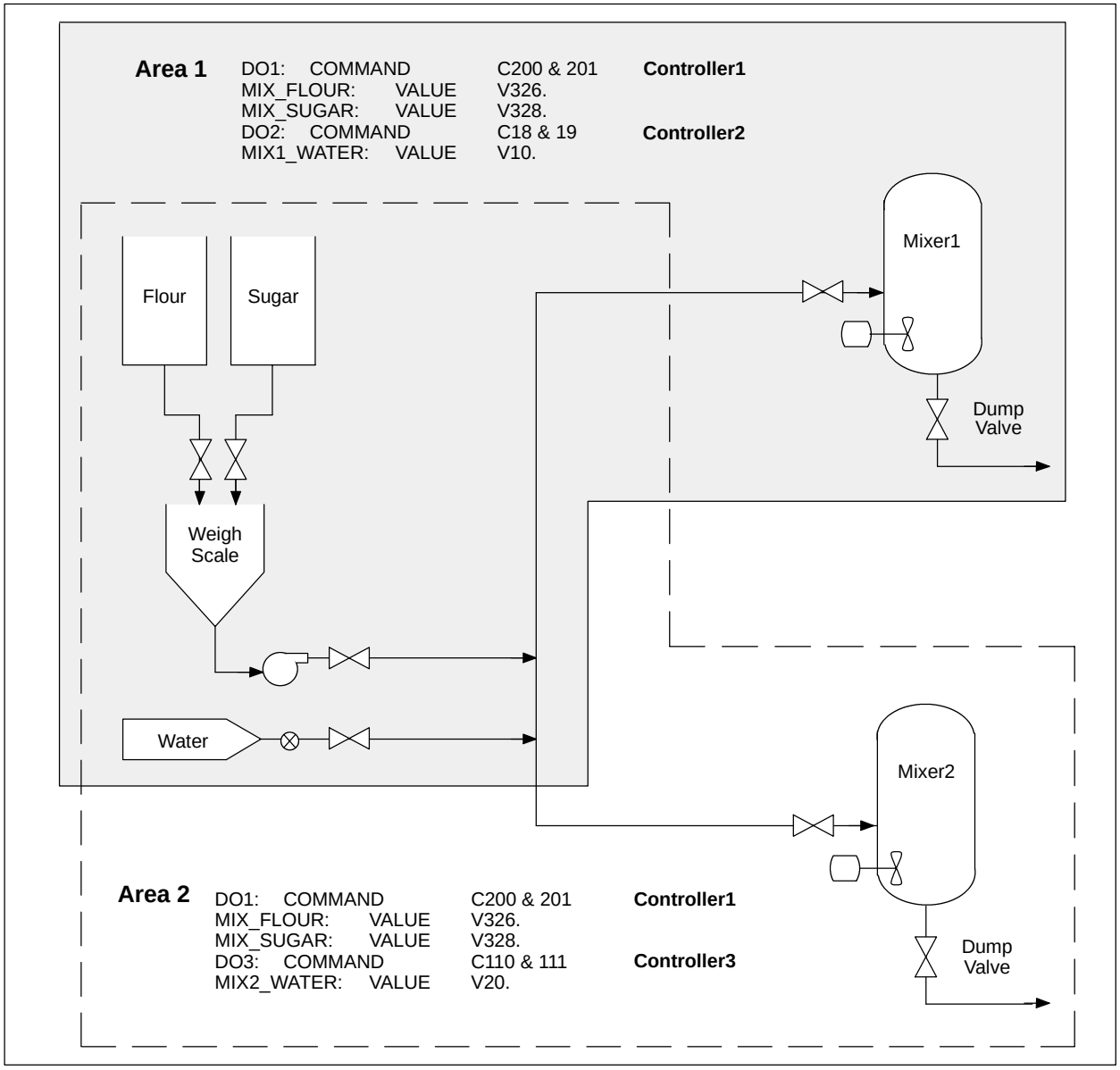


Figure 1-12 Recipe Areas

The third step in recipe configuration is to name the recipe and link it to a recipe area. Before you can link a recipe to the recipe area, you must create the recipe.

1. Select **Data->Recipe** from the menu bar. The Recipe Configuration Menu display appears (Figure 1-13).

2. Select the **Add** pushbutton from the Recipe Configuration Menu display. The Add Recipe dialog box appears (Figure 1-13).

3. Enter a name for the recipe or subrecipe. The space, the character combinations -> and <- and the characters ; \ , " are invalid for recipe names.

4. Enter a description for the recipe or subrecipe. The characters ; \ " are invalid for recipe descriptions.

5. Select a recipe type: Recipe or Subrecipe.

6. Select **OK**. OSx adds the new recipe name to the Recipe Configuration Menu and exits the display.

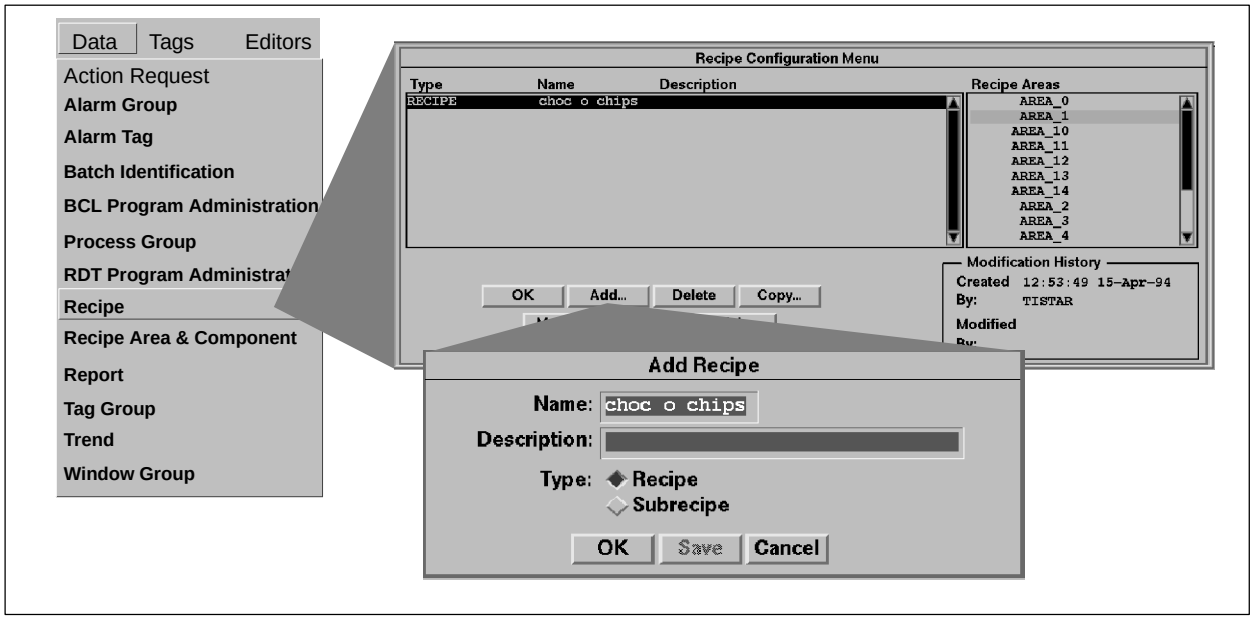
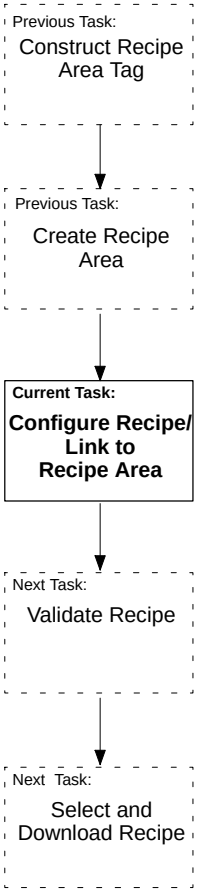


Figure 1-13 Linking Recipes and Recipe Areas



To link the recipe to the recipe area, follow these steps:

1. Select the appropriate recipe and recipe area from the Recipe Configuration Menu display. When you have successfully linked a recipe selection with a recipe area the associated recipe area is shown in yellow. If the selected recipe has been linked to a recipe area and validated, the associated recipe area is white. You do not have to link the recipe and the recipe area to build the recipe, only to validate it.

2. Click the **Modify** pushbutton. The Recipe Configuration Editor display appears (Figure 1-14).

3. For each recipe component, enter data type, low/high ranges, initial value, units, and whether to allow scaling.

4. Select **Save**. OSx saves your recipe.

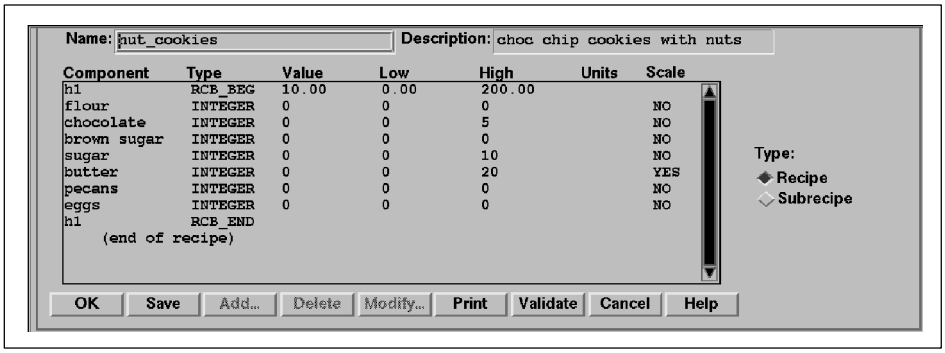
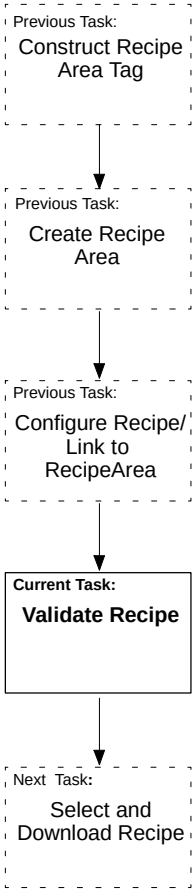


Figure 1-14 Recipe Configuration Editor Display



After you have entered all the recipe components and saved your work, the next step is to validate the recipe. Select the **Validate** pushbutton from the Recipe Configuration Editor display. The system compares the components of the current recipe with the components of all linked recipe areas. Validation ensures compatibility between the recipe and any associated recipe area (Figure 1-15).

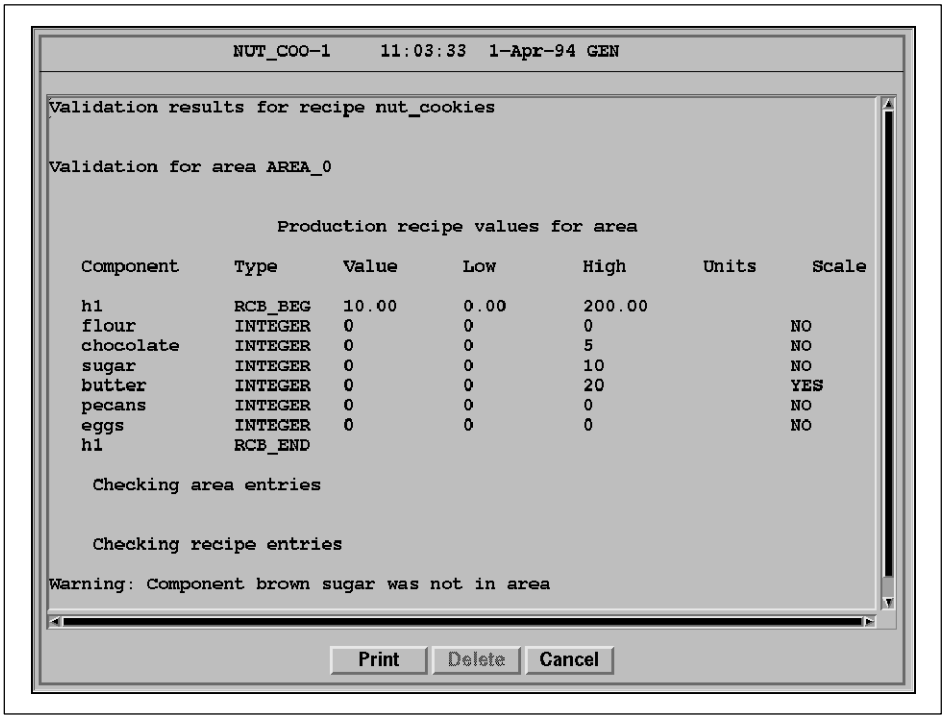
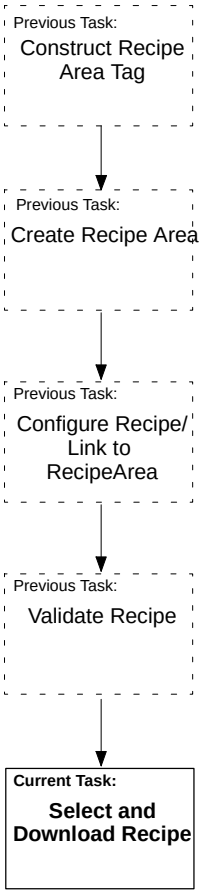

NUT_COO-1 11:03:33 1-Apr-94 GEN
Validation results for recipe nut_cookies
Validation for area AREA_0
Production recipe values for area
Component Type Value Low High Units Scale
h1 RCB_BEG 10.00 0.00 200.00
flour INTEGER 0 0 0 NO
chocolate INTEGER 0 0 5 NO
sugar INTEGER 0 0 10 NO
butter INTEGER 0 0 20 YES
pecans INTEGER 0 0 0 NO
eggs INTEGER 0 0 0 NO
h1 RCB_END
Checking area entries
Checking recipe entries
Warning: Component brown sugar was not in area
Print
Delete
Cancel

Figure 1-15 Recipe Validation Display



The final step for configuring a recipe is to select and download the recipe to the recipe area. If necessary, you can modify values and change the scaling option before downloading the recipe (Figure 1-16). Refer to Chapter 7 for information on both recipe selection and downloading a recipe. Figure 1-17 shows a sample recipe downloading to a recipe area.

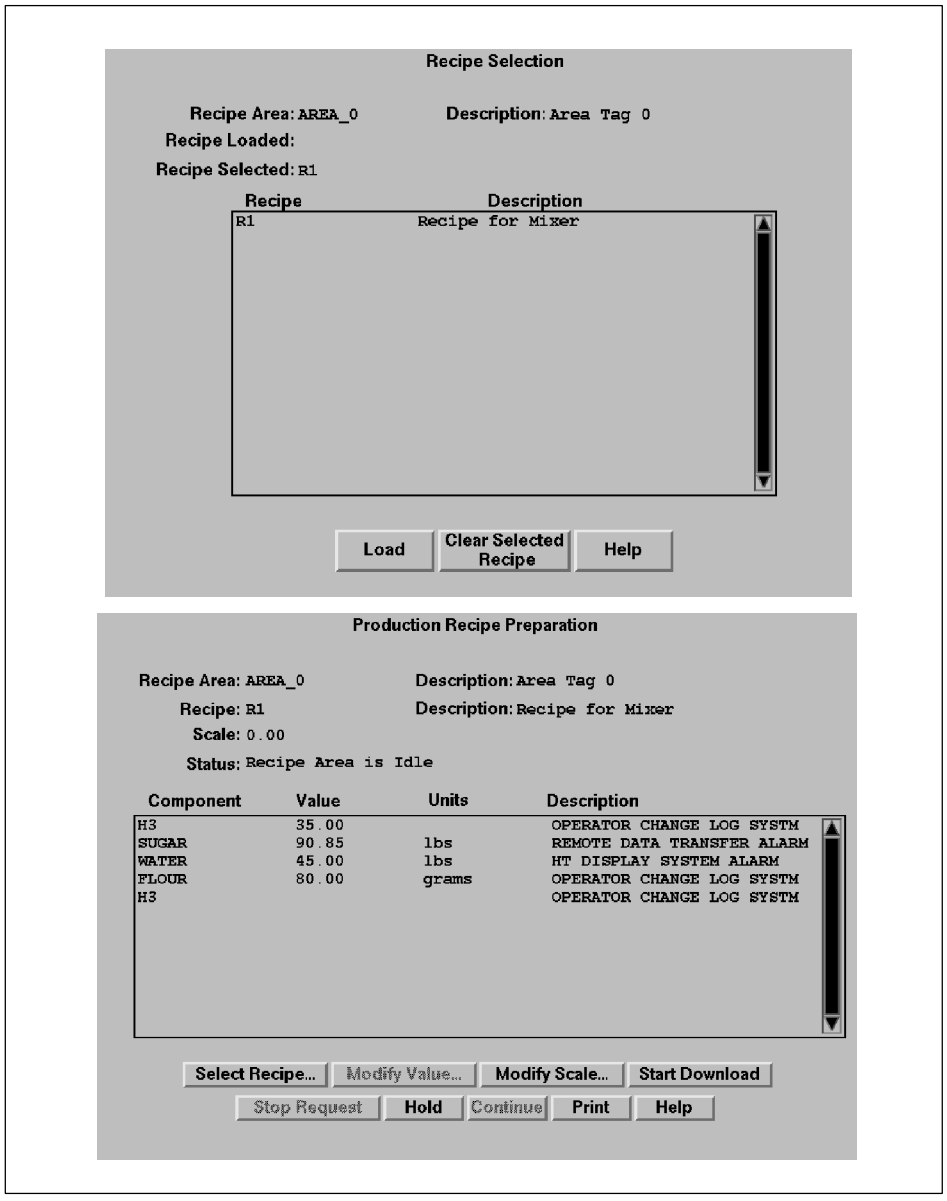


Figure 1-16 Recipe Selection/Production Recipe Preparation Screens

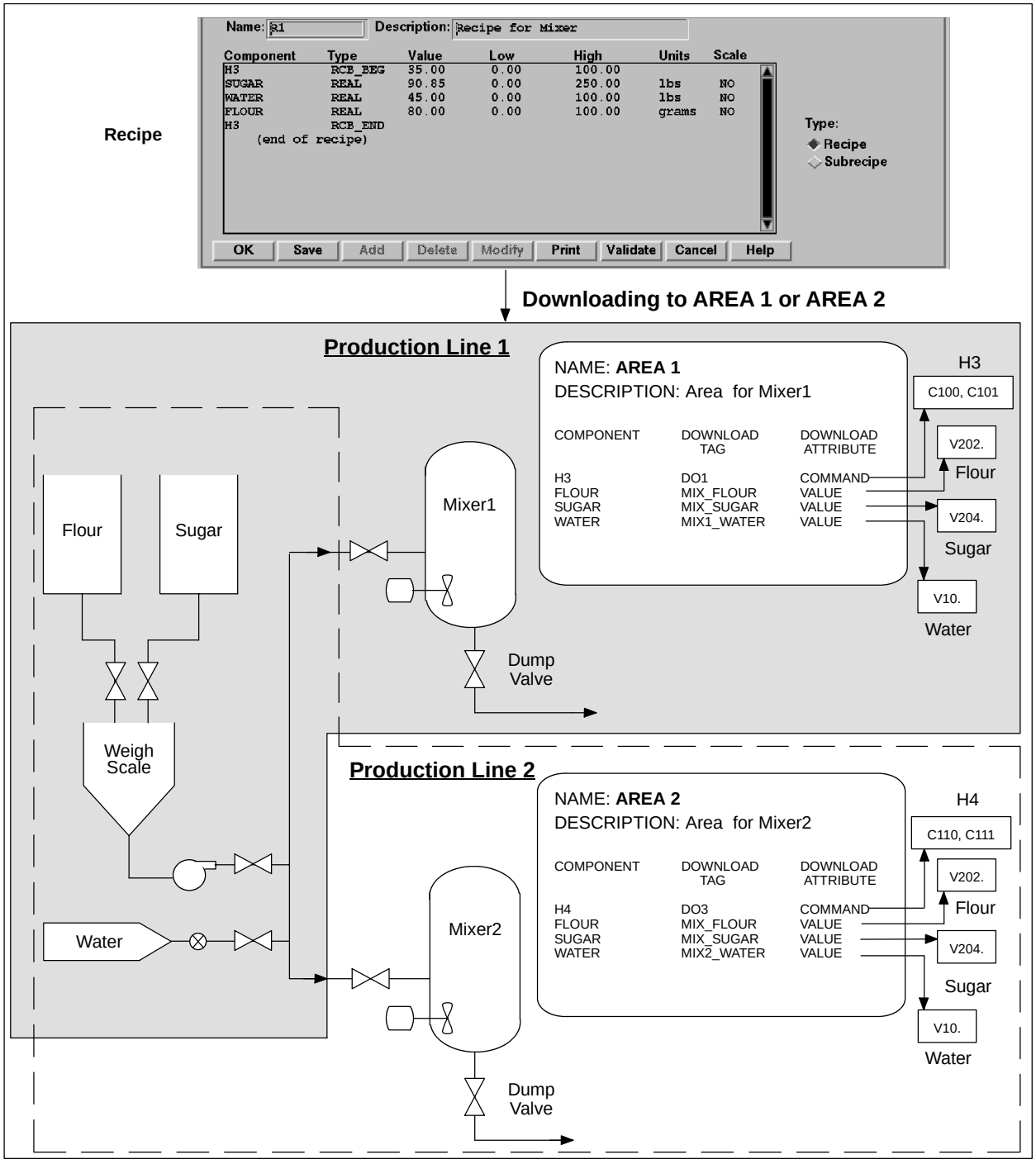


Figure 1-17 Sample Recipe and Recipe Areas

Chapter 2

Defining Area Tags

2.1 Overview 2-2

2.2 Choosing Area Tag Data 2-3

2.3 Defining Tag Attribute Data 2-4

2.4 Deleting an Area Tag 2-5

2.1 Overview

Use area tags to control and provide information about recipes that have been downloaded as production recipes to a particular area. These tags provide status information about recipe activity, alarms, handshaking, and administrative information for recipe areas.

The system default allows you to configure up to 50 area tags that represent the locations receiving recipe values. However, if you resize the tag capacity of your system, you can actually configure up to 200 area tags. Before you start, you must configure the control nodes. If necessary, adjust the maximum number of records and change the default deadband values for tag types that you use.

Area tags are created by the same method that you use to create the other process I/O tags. You can choose from these methods of configuring tags:

Use APT Create the application program using APT. Then mark the names that you assigned to APT program objects for translation into tags.

Use the Tag Configurator Select **Tags->Configurator** from the main menu bar. Enter the tag information and save.

Use a Spreadsheet Program Create the application program using a spreadsheet. Enter tag information into the spreadsheet and incorporate it into the OSx database. The spreadsheet must allow you to save your data in an ASCII format in which the variables are separated by commas.

Use any ASCII Text Editor Create an ASCII file, enter tag information into the file, and incorporate it into the OSx database.

The *SIMATIC PCS 7 OSx Process Configuration Manual* chapter on configuring tags describes these methods. For APT, refer to the *SIMATIC APT Programming Reference (Tables) Manual*.

2.2 Choosing Area Tag Data

When you define an area tag, you need to enter the types of information shown in Table 2-1.

Table 2-1 Tag Header Fields

Field	Definition
Record	Identifies line of tag information. Enter `T` for a tag identifier line. Enter `A` for a tag attribute line.
Control Node	Enter the name of the target controller, e.g., SIMATIC 555, S5 (up to 12 characters).
Tag Type	Enter `area` for the tag type.
Tag	Enter the name of the tag (up to 12 characters). All characters are valid except the comma, the double quote, and the space. The tag name must begin with a character (A-Z, a-z).
Description	Enter the description of the tag (up to 30 characters).
Process Group	Enter the process group number indicated with a 32-bit hex number. Each bit corresponds to one of 32 process groups. For each process group with which the tag is to be associated, its corresponding bit is set to 1. Refer to the *SIMATIC PCS 7 OSx Process Configuration Manual* for the format.
Manual Set	Indicates Y(es) or N(o) that you can enter data manually for the tag for trending purposes. Only applies to users of the historical trend utility (htu).
Parent	If you have structured your application program into units, tags within a unit can all be associated by means of a parent tag. Assign the same parent tag name to tags that pertain to the same unit. The name of a parent tag contains up to 12 characters.
Attribute	Enter the tag attribute, e.g., `status, hold_req, or scale factor`. See Section 2.3.
Memory	Enter the memory location in the controller where the tag attribute data is located, e.g., LPV1, V55, WY41. Refer to the controller documentation and the *SIMATIC PCS 7 OSx Process Configuration Manual* appendix on Controller Memory Types for details about memory types.
Locations	Enter the number of memory locations required for an attribute. This information varies with the tag attribute. Refer to your controller documentation for details about the number of memory locations required by each attribute.
Upload	Enter `Y`(es) or `N`(o) for a range or limit or any other data value that is required for use in the control node Tag Download command. This command is used to restore current values in the controller when you have reloaded or restarted the controller program. Since upload values typically do not change, operations stations collect them from the network when required (only on every other event scan period).
Twenty %	Enter `Y`(es) if 20% offset is done on analog values. A span of 0 to 5.0 volts (0 to 20.0 milliamps) is referred to as a span of 0% to 100%. A span of 1.0 to 5.0 volts (4 to 20.0 milliamps) is referred to as a span of 20% to 100%.
Autolog	Enter `Y`(es) if you want a log message to be generated when this attribute value changes in the OSx database.
InitValue	Enter the initial value assigned to the attribute.

2.3 Defining Tag Attribute Data

You need to define the following tag attributes.

Hold_req The hold_req attribute allows a controller to request that a recipe area be placed in a Hold state (suspends downloading of recipes) or a Continue state (allows downloading of recipe). This can be a network-scanned location.

The recipe download can only be suspended on a **RCB_BEG** block. (See Chapter 4 for a full description of Recipe Control Blocks.) If you request a hold before the download starts, the download is suspended when the next **RCB_BEG** block appears. If you request a hold during an active download, the hold request is not recognized until the system evaluates a **RCB_BEG** block and waits for the In_use bit to become available.

Scale_factor The scale_factor attribute is a floating point value that can be uploaded or downloaded to a controller and can be scanned. If, during recipe configuration, the scale flag is set to **YES** for integer or real-type recipe components, those recipe values are multiplied by this factor when the production recipe is created.

Scale_high and scale_low The scale_high and scale_low attributes are floating point values that define the upper and lower limits for the scale factor of the recipe area.

Plc_rcp_req The plc_rcp_req attribute is a signed integer value that allows the controller to control recipe-download activity. A positive integer value activates a request for recipe download. OSx appends the positive integer value to an R prefix to produce the requested recipe name Rn. OSx then searches for a configured recipe of the name Rn to attempt the download. A negative integer value of –3 activates a download-only request of a previously selected recipe. A value of –1 activates a request to abort a recipe-download request. The default state is –2, no action.

Status The status attribute is generated in OSx to show recipe area activity, and is read-only. OSx reserves the three least significant bits for alarming and the communication subsystem. The Active bit shows the status of network communication for a tag, and the Alarm UnAck bit shows the alarms that have not been acknowledged for a point. The Manual Set bit allows you to enter data manually for the tag for trending purposes. This applies only to users of the historical trend utility (htu). The remaining thirteen bits are set in OSx. For a description of each bit, see Appendix A, Recipe Area Table.

2.4 Deleting an Area Tag

You can delete an area tag that you have incorporated in the OSx database. However, if the area tag currently has a production recipe, you must remove its recipe from production before you can delete the tag. See the procedure below.

You can also delete a tag that is a component of a recipe area. When you delete the tag the system automatically removes the component from the recipe area. However, if the tag is a component of a recipe area that currently has a production recipe, you must remove its recipe from production before you can delete the tag. See the procedure below.

To remove a recipe from production, follow these steps.

1. Set the system to the Operate state. The navigation area appears.
2. Select **Recipe** from the Directory in the navigation area.
3. Select the Recipe Area that you want to delete or the recipe area that contains the tag you want to delete.
4. Click the **Select Recipe** button. This removes the recipe from production. Now you can delete the tag from the SIMATIC PCS 7 OSx system.

Refer to your APT documentation or the *SIMATIC PCS 7 OSx Process Configuration Manual* for additional information on deleting tags.

Chapter 3

Defining Recipe Areas

3.1 Recipe Area Configuration Overview 3-2
Identifying Recipe Areas 3-2
Preparing to Configure Recipe Areas 3-2
Configuring on Two Stations 3-3

3.2 Configuring Recipe Areas 3-4
Entering the Area Name 3-4
Adding a Component 3-4
Modifying Components, Downloading Tags and Attributes 3-6

3.1 Recipe Area Configuration Overview

Identifying Recipe Areas

When you configure recipe areas, you identify sets of memory locations in the controllers for downloading recipe values. Tags and attributes identify these locations. When you download the recipe for production, the system copies the recipe values to the memory locations identified by the attributes.

The system default allows you to configure up to 50 recipe areas. However, if you resize the tag capacity of your system, you can actually configure up to 200 recipe areas.

Depending on how you write the program in the controller, you can associate a recipe area with one or more production lines.

Preparing to Configure Recipe Areas

Before you configure the recipe areas, configure the area tags and all tags used for recipe values and identify the recipe area components. After you configure the recipe areas, you can configure recipes, link them to recipe areas, and validate them.

To access the Recipe Area and Component Configuration display, select **Data->Recipe Area & Component** from the menu bar (Figure 3-1). Field definitions are described below.

Recipe Area Name The Recipe Area Name field contains the tag name that identifies the recipe area (up to twelve characters). The space, the character combinations -> and <- and the characters ; , \ ” are invalid for a recipe area name.

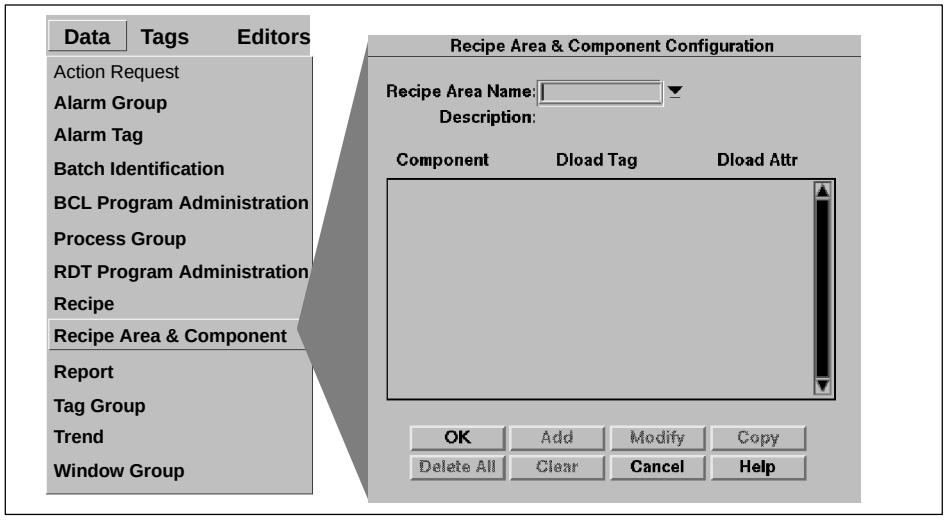


Figure 3-1 Recipe Area and Component Configuration Display

Description The Description field is a character string (up to 30 characters) that describes the function, the purpose, or the location of a recipe area. If you used the description field to create an area tag, the description field automatically fills when you enter a recipe area name. The characters ; \ ” are invalid for descriptions.

Component The Component field lists all components that can be included in recipes associated with the recipe area. The components are listed alphabetically after the recipe area is saved.

Dload Tag The Dload Tag field describes a tag that contains an attribute that receives the download value for the component.

Dload Attr The Dload Attribute field describes the attribute that receives the download value for the component.

The pushbutton selections, located at the bottom of the Recipe Area and Component Configuration display, are defined below.

Add The Add option allows you to add a component that consists of a component name, download tag, and download attribute.

Modify The Modify option allows you to modify an existing component that consists of a component name, download tag, and download attribute.

Copy The Copy option allows you to copy the current component list to another designated recipe area.

Delete All The Delete All option allows you to delete the current component list for the recipe area.

Clear The Clear option allows you to clear all current recipe area information from the screen.

Configuring on Two Stations

It is possible for one user to start Recipe Configuration on one station and for another user to start Recipe Area and Component Configuration on another station in the system at the same time. When this happens, nothing prevents the users from working on the same recipe area at the same time.

Be aware that changes made in one configuration utility (for example, new components added to an area in Recipe Area and Component Configuration) do not show up in the other one (for example, the select list of components for an area in the Add Component Values area of Recipe Configuration) until the users exit and reenter.

3.2 Configuring Recipe Areas

Entering the Area Name

The recipe area name must match the name of the area tag that you assigned during tag configuration. If you enter a name that is not yet installed in OSx, or the name of a non-area tag, the system returns an error message.

To enter the recipe area name from the Recipe Area and Component Configuration display follow these steps.

1. Click the Recipe Area Name field.
2. Either enter the appropriate area tag name, or click the right mouse button to choose one from the pop-up list. The entry is not case-sensitive (upper and lower case are both valid).

If you enter the name of a previously configured recipe area, the system fills in all of the associated information. Components appear in alphabetical order. You can then view or modify the recipe area data.

Adding a Component

For each recipe area that you configure, you enter component names to represent all the tags and associated attributes in the recipe area. The component name more clearly describes the function or purpose of a tag than does the tag ID. You can use any recipe area components in one or more recipes. Among other things, these components can represent any of the following elements.

- Ingredients, such as flour, water, or sugar
- Processes, such as cleaning equipment, or discharging product
- Processing parameters, such as mix times or temperatures
- Recipe controls, such as subrecipes and recipe control blocks
- Flags to indicate which control logic to use

You must associate every recipe area component with a download tag and attribute. The tag and attribute identify the database location and the optional controller location to which you download the value of the associated recipe component. You can use any tag except for TEXT, DO10, and DI10. Tags do not have to be located in the same controller.

NOTE: High Range (H_RANGE) and Low Range (L_RANGE) are not valid attribute selections. If you try to enter these values, an error condition results during recipe download that aborts the download operation. Do not use read-only attributes.

To add components from the Recipe Area and Component Configuration display, follow these steps.

1. Select a recipe area from the Recipe Area and Component Configuration display.
2. Click the **Add** button. The Add Recipe Component dialog box appears (Figure 3-2).
3. Enter the component name and press **Enter**.

   If you later change the component name, you will also need to reenter the tag and attribute information.
4. Either enter the name of the tag you want to associate with the component in the appropriate field, or click the button to the right of the Download Tag field with the left mouse button to select from a list of download tags.
5. Either enter the name of the attribute associated with the download tag in the appropriate field, or click the button to the right of the Download Attribute field with the left mouse button to select from a list of download attributes.
6. Select **Save**. When you save a recipe area, the components are sorted in alphabetical order.

NOTE: When you configure a tag for a recipe, do not configure it for use in an action request, since tag values can be changed during action requests.

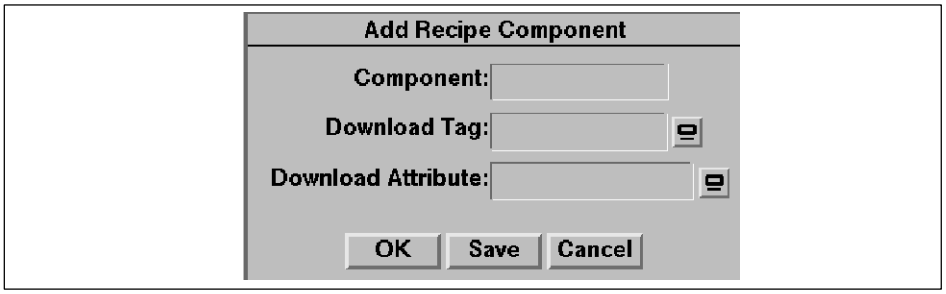


Figure 3-2 Add Recipe Component Dialog Box

NOTE: If the same component is listed in more than one recipe area, it must be associated with download tags and attributes of the same data type and engineering units; otherwise, the value interpretation is not consistent.

Modifying Components, Downloading Tags and Attributes

To modify components from the Recipe Area and Component Configuration display, follow these steps.

1. Select a component from the component list. Click the **Modify** button. The Modify Recipe Component dialog box appears (Figure 3-3).
2. Enter the component name you want to modify.
3. Enter the download tag name you want to modify. Click the button to the right of the Download Tag field with the left mouse button to display a select list of download tags.
4. Enter the download attribute name you want to modify. Click the button to the right of the Download Attribute field with the left mouse button to display a select list of download attributes.

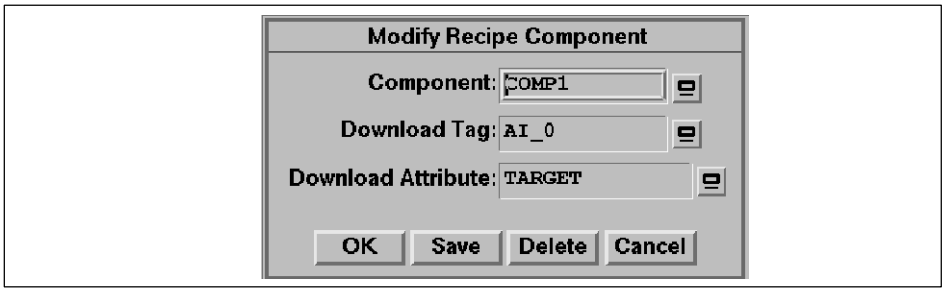


Figure 3-3 Modify Recipe Component Dialog Box

Chapter 4

Using Recipe Control Mechanisms

4.1 Recipe Control Mechanisms 4-2
Using Subrecipes for Control 4-2
Using Recipe Control Blocks (RCBs) 4-4
RCB_WAIT 4-4
RCB_BEG and RCB_END 4-5

4.2 Using RCBs to Control Recipe Downloads 4-6
Nesting Recipe Control Blocks 4-6
Handshake Communication 4-6
Phased Download 4-10

4.1 Recipe Control Mechanisms

Using Subrecipes for Control

Subrecipes contain one or more components in common with various recipes. Recipes can contain one or more subrecipes.

The last value listed for a common component takes precedence when the recipe is downloaded. Therefore, if a subrecipe contains components that duplicate one or more components in the recipe, the position of the subrecipe is important. Refer to Section 5.3 for detailed information on adding components and component values to recipes and subrecipes.

When you use subrecipes, observe the following rules.

- A recipe can contain one or more subrecipes.
- A subrecipe cannot contain another subrecipe.

Figure 4-1 shows an example of recipe-to-subrecipe organization. The subrecipe is a raw dough for producing cookies. Recipe **CHOC_CHIPS** (Figure 4-1) refines the raw dough with vanilla, sugar, and chocolate chips. Sugar is a component common to both the subrecipe and the recipe. By adjusting the position of the subrecipe in the list of recipe components for chocolate chip cookies, you can decrease or increase the sweetness of the cookies.

In the example, subrecipe **RAW_DOUGH** contains more sugar (90 lbs) and is listed first. The chocolate chip recipe value for sugar (75 lbs) is listed last. Since the last value listed for duplicate components takes precedence, the production recipe (**CHOC_CHIPS**) contains less sugar.

If the example listed the subrecipe (**RAW_DOUGH**) after the recipe entry for sugar (75 lbs), then the subrecipe's sugar value would take precedence and recipe **CHOC_CHIPS** would contain a sugar value of 90 lbs.

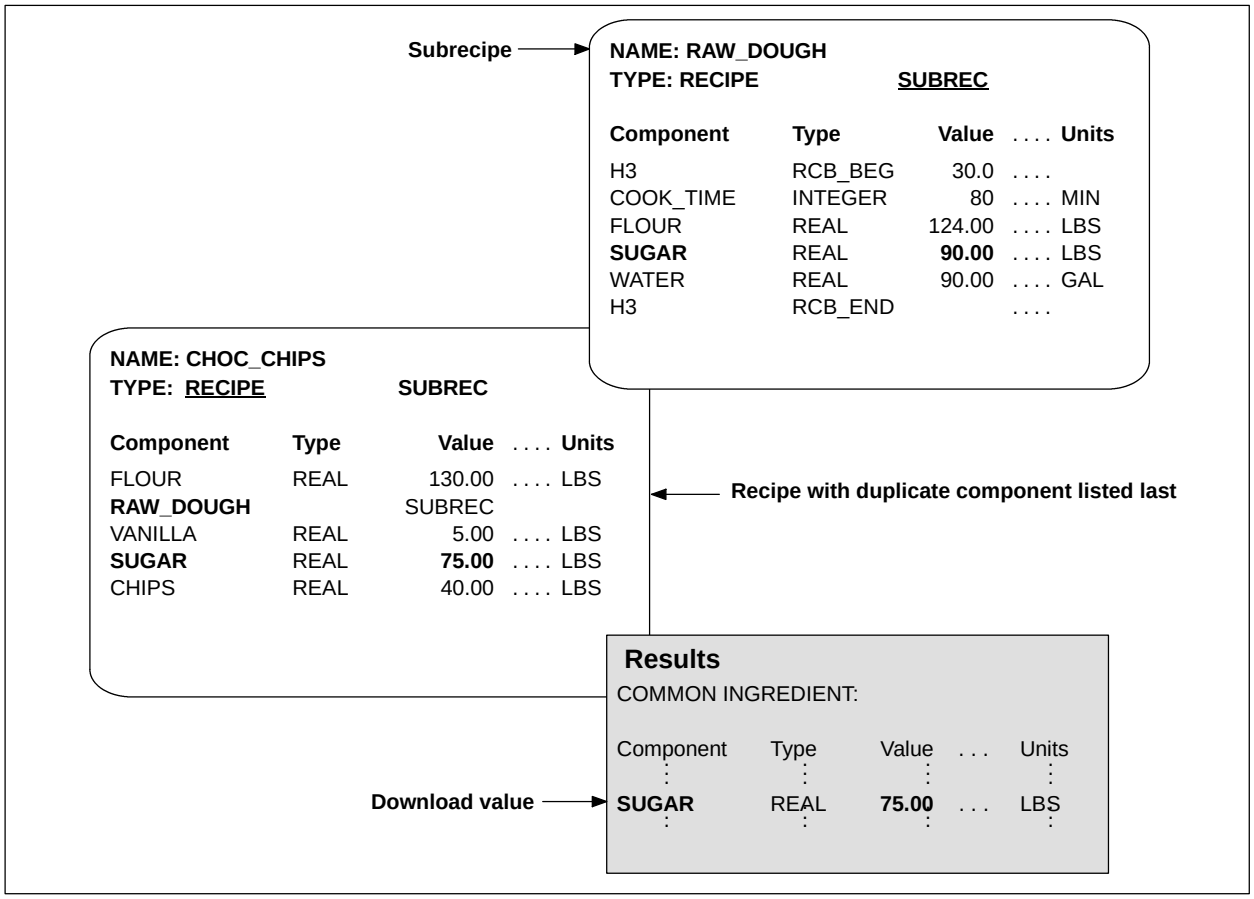


Figure 4-1 Recipe-to-Subrecipe Organization Example

Using Recipe Control Blocks (RCBs)

SIMATIC PCS 7 OSx provides three recipe control block statements (RCBs) that you can use together to produce a more complex recipe control strategy.

- **RCB_BEG** and **RCB_END** are used together in a recipe or subrecipe to form a control block.
- **RCB_WAIT** controls the order that the recipe components are downloaded to the controller.

When OSx downloads a recipe, the components are sent in indeterminate order unless you include recipe control block statements in the recipes. These statements give you some control over the download process.

RCB_WAIT

RCB_WAIT is a recipe control block statement for controlling the order that the recipe components are downloaded. Use **RCB_WAIT** if you want OSx to download certain recipe component values before others. OSx sends all the recipe values listed before an **RCB_WAIT** first; when the controller has received the values, the download continues. Using **RCB_WAIT** does not require that you define a tag for the **RCB_WAIT**; it is only for procedural control (Figure 4-2). A recipe can contain more than one **RCB_WAIT**.

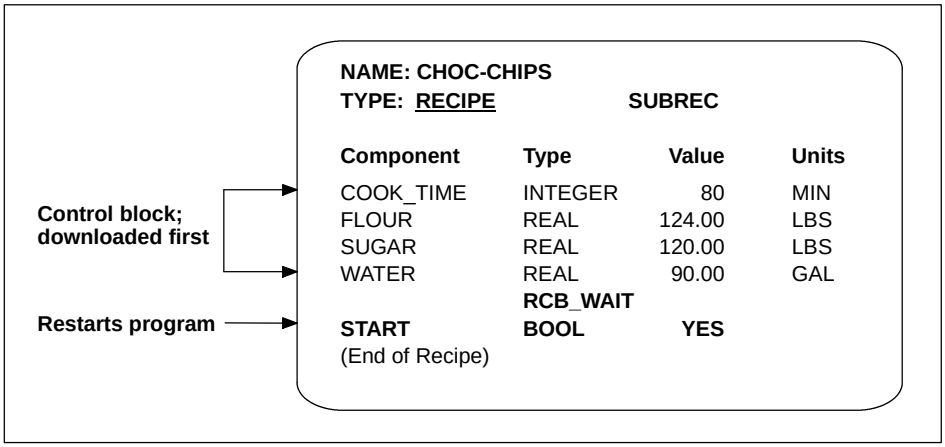


Figure 4-2 Recipe Using RCB_WAIT

In Figure 4-2, the controller receives all the component values listed before the **RCB_WAIT**; OSx then sends the remaining component values. **START** is a simple boolean value that maps to a control relay to cause production of the product to begin.

RCB_BEG and RCB_END

RCB_BEG and **RCB_END** are component types that are used together in a recipe or subrecipe to form a control block.

In Figure 4-3, OSx ensures that all recipe values listed before **RCB_BEG** are sent to and received by the controller first. When the recipe management package encounters the value for **RCB_BEG**, it uses the **RCB_BEG** to activate a handshake procedure with the controller. Use a recipe control block to suspend or cancel the download while the controller is locking out the download area.

When you use a recipe control block, observe the following rules.

- Use the same component name for **RCB_BEG** and its associated **RCB_END**. (In the example, both **RCB_BEG** and **RCB_END** have the component name **H3**.)
- Place **RCB_BEG** before **RCB_END** in the component list.
- Enter a time-out value for **RCB_BEG**. Time-out defines the number of seconds that OSx waits before abandoning the download attempt.

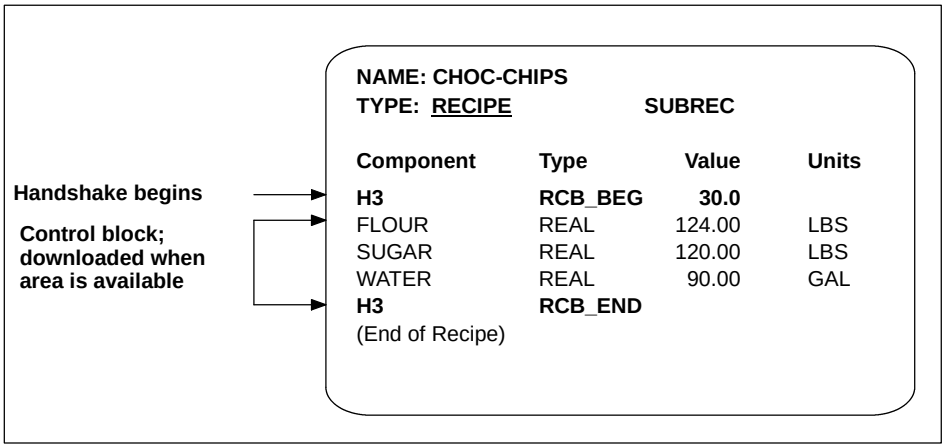


Figure 4-3 Recipe with RCB_BEG and RCB_END

4.2 Using RCBs to Control Recipe Downloads

Nesting Recipe Control Blocks

You can use more than one control block in a recipe. Nest control blocks according to the guidelines in Figure 4-4.

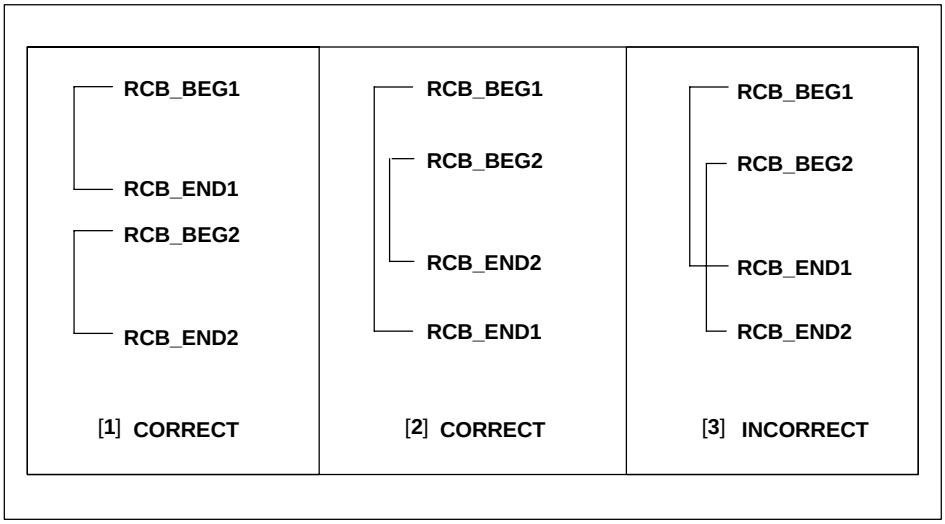


Figure 4-4 Examples of Valid and Invalid Nesting of RCBs

Figure 4-4 shows that you can use control blocks sequentially within a recipe as in example 1, or you can place one control block embedded within another as in example 2. In embedded nesting, you must use a different handshaking component for each recipe control block. Control blocks cannot overlap, as example 3 shows.

Handshake Communication

When you download a recipe containing **RCB_BEG**, the system performs a handshake procedure. (You assigned the tag used for handshake communication when you configured the recipe area component for the **RCB_BEG**.) The tag must be an unsigned integer that has the two most significant bits available for recipe communication. Use a digital output tag and command attribute for this purpose (Figure 4-5). The example uses C-memory for the handshake tag; you must configure the handshake bits to the tag attribute in the OSx database. In Figure 4-5, the first of two memory locations is specified as C101.

The program in the controller sets the In_Use bit to indicate if the recipe area is available or busy. You can program the controller to begin production after all recipe values download successfully. The recipe management software sets both the In_Use and Stable bits to 1 after all the recipe values download.

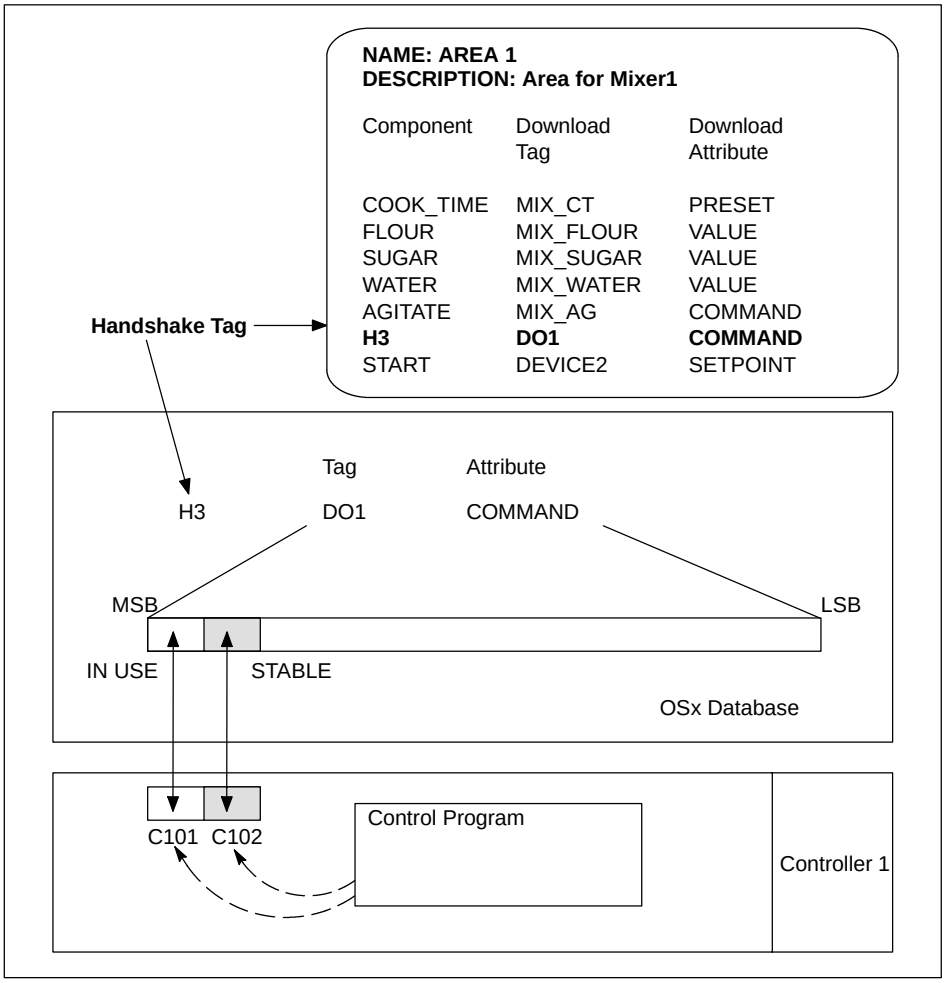


Figure 4-5 Handshake Tag and Communication Bits

Except in situations where the recipe download aborts, the program in the controller must also signal the availability of the recipe area by setting the In_Use bit to 0 after the program finishes using the recipe values.

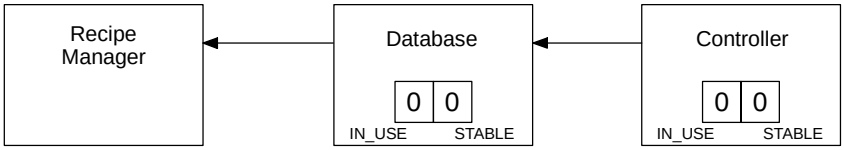
⚠ WARNING

Failure to correctly interlock the recipe download communications can cause unpredictable controller operations that could result in death or serious injury to personnel, and/or damage to equipment.

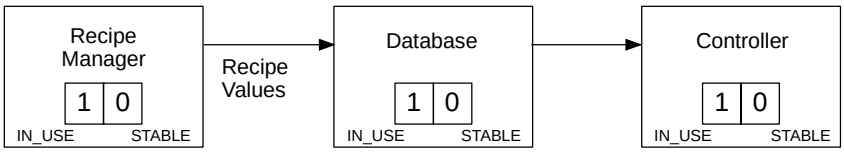
Be sure that you understand proper nesting and handshaking techniques so that production does not begin until all recipe values have been downloaded successfully.

The following steps show the handshaking required for downloading recipe values.

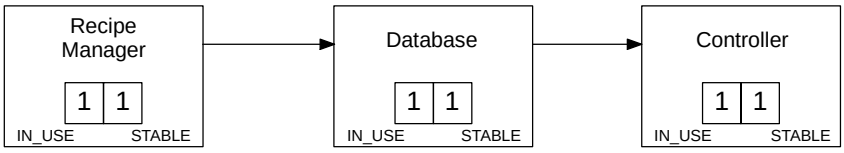
1. When OSx encounters an **RCB_BEG** block, the system reads the value of the control bits (In_Use and Stable) from the database. If the In_Use bit is set to 1, the recipe area is currently in use (owned by another controller), and the recipe values cannot be downloaded. An internal time-out begins; if the In_Use bit does not change to zero before time-out occurs, the recipe download aborts.



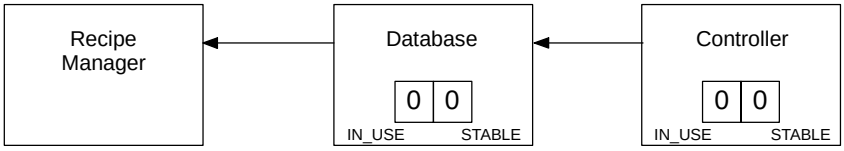
2. If the In_Use bit is 0, the recipe management software sets the In_Use bit, clears the Stable bits in the database, and starts to download recipe values. (Setting the In_Use bit declares ownership of the recipe area, and clearing the Stable bit informs the controller that recipe values were not downloaded.)



3. After the recipe values download, OSx sets the Stable bit in the database and controller. Setting both the In_Use and Stable bits to 1 informs the controller that all recipe values were sent and are available for the controller control program to use.



4. After the controller program executes and the controller is ready for a new recipe (or recipe control block), the control program in the controller clears the control bits.



Phased Download

OSx allows you to download a recipe in stages, a few components at a time, using both multiple control blocks and duplicate components within a recipe. Unlike the relationship between recipes and subrecipes, the duplicate components in different control blocks retain their own values.

Recipes and subrecipes can contain more than one **RCB_BEG** and **RCB_END** control block. Within any given recipe control block, component names are unique; however, you can reuse component names outside the control block. When a recipe contains duplicate components within the same recipe control block, the last value listed in the control block, either by subrecipe expansion or direct entry, is the value used in the production recipe.

When you use sequential nesting, as in Figure 4-6, different control blocks within a recipe can use the same tag for handshake communication. This option serves as a control trigger between the phases, or control blocks. When the recipe downloads, the first control block waits up to 30 seconds for an available designated recipe area. When the area is ready, the values for cook time (80 minutes), flour (124.0 pounds), and sugar (120.0 pounds) are sent to the controller.

When you use embedded nesting (as in example 2 in Figure 4-4 on page 4-6), you must use a different handshaking tag for each recipe control block.

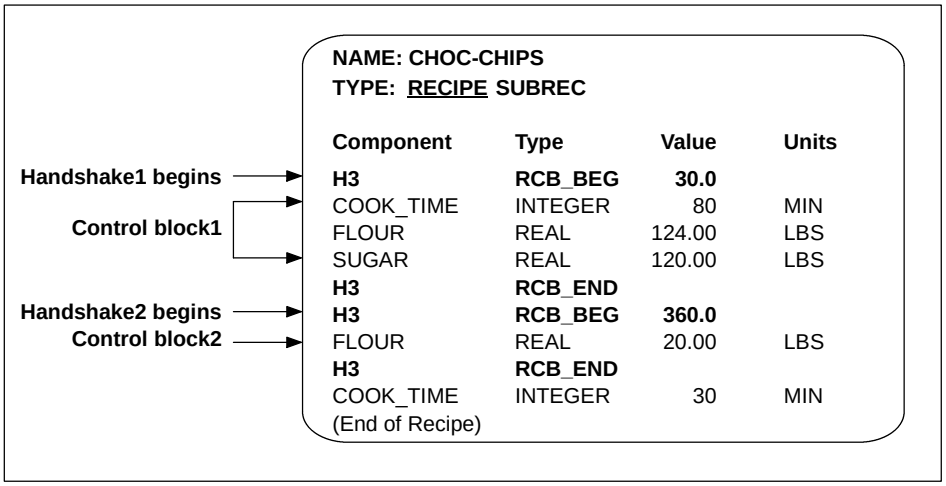


Figure 4-6 Using Control Blocks with Duplicate Components

After receiving the first control block, the recipe attempts to continue the download. The H3 tag has the In_Use and Stable bit set, preventing the second phase from sending its values. This control block waits up to 360.0 seconds for an available recipe area. The controller program can then operate on the phase 1 values until the controller is ready to receive new values, at which time the controller clears the H3 In_Use bit. When the area is ready, the download continues with another value for flour (20.0 pounds). Finally, the download concludes with another cook time (30 minutes).

Use the control block to ensure that necessary values are available before the dependent values use them. Figure 4-7 shows a typical application. This recipe requires setting both the **MIX_TIME** and **MIX_AMOUNT** values before the mix operation starts. Use the control block to ensure that these values change before the mix operation starts.

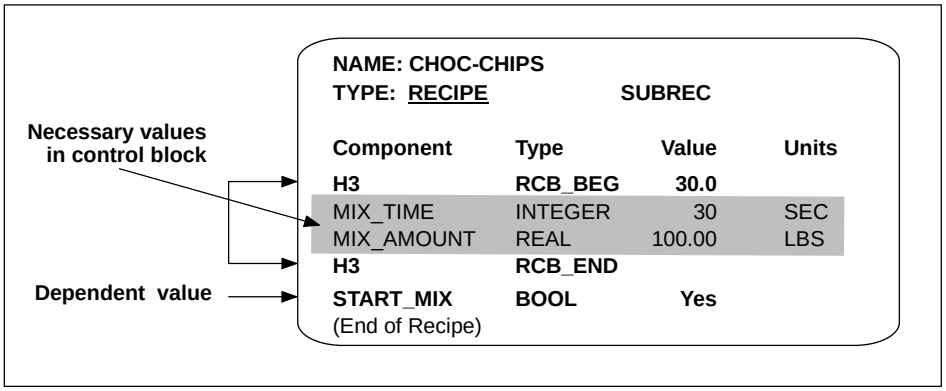


Figure 4-7 Setting Necessary Values in a Control Block

Chapter 5

Defining Recipes

5.1 Recipe Configuration Overview 5-2
Overview 5-2
Accessing the Recipe Configuration Menu Display 5-2

5.2 Using the Recipe Configuration Menu Display 5-4
Selecting an Option 5-4
Specifying the Recipe/Subrecipe Name 5-4
Specifying the Entry Type 5-5
Specifying the Recipe/Subrecipe Description 5-5
Adding a Recipe/Subrecipe 5-5
Deleting a Recipe/Subrecipe 5-6
Copying a Recipe/Subrecipe 5-6
Modifying a Recipe/Subrecipe 5-8

5.3 Configuring Recipes 5-10
Adding Components 5-10
Choosing the Data Type 5-12
Entering Range Limits 5-13
Entering the Component Value 5-13
Entering Units 5-13
Setting the Scale Flag 5-13
Modifying Component Values 5-14
Linking Recipes and Recipe Areas 5-15

5.1 Recipe Configuration Overview

Overview

The SIMATIC PCS 7 OSx recipe system allows you to configure up to a maximum of 500 recipes. You can configure recipes in either the Offline or Operate state. To configure recipes, follow this process.

1. Add the recipe/subrecipe name, type, and description to the system.
2. Define the recipe/subrecipe components and parameters.

Use the Recipe Configuration Menu to add, modify, and define recipes and subrecipes.

Accessing the Recipe Configuration Menu Display

Before you define the contents of a recipe or subrecipe, you must add the recipe name and description to the Recipe Configuration Menu.

To access the Recipe Configuration Menu display, select **Data->Recipe** from the menu bar (Figure 5-1). The Recipe Configuration Menu display fields are described below.

Type The Type field indicates either a recipe or subrecipe.

Name The Name field provides a simple means of referencing the recipe or subrecipe. The space, the character combinations -> and <- and the characters ; , \ ” are invalid for recipe names.

Description The Description field displays a meaningful definition of the recipe or subrecipe. The characters ; \ ” are invalid for descriptions.

Recipe Areas The Recipe Areas field lists all recipe area names configured in the system. If the current recipe selection has been linked to a recipe area and validated, the associated recipe area shows in white. If the selected recipe has been linked to a recipe area but not validated, the associated recipe area shows in yellow.

Modification History The Modification History field displays the time and date a recipe was created or modified as well as the User ID of the person who created or modified the recipe.

Figure 5-1 shows how to access the Recipe Configuration Menu display from **Data->Recipe** on the menu bar.

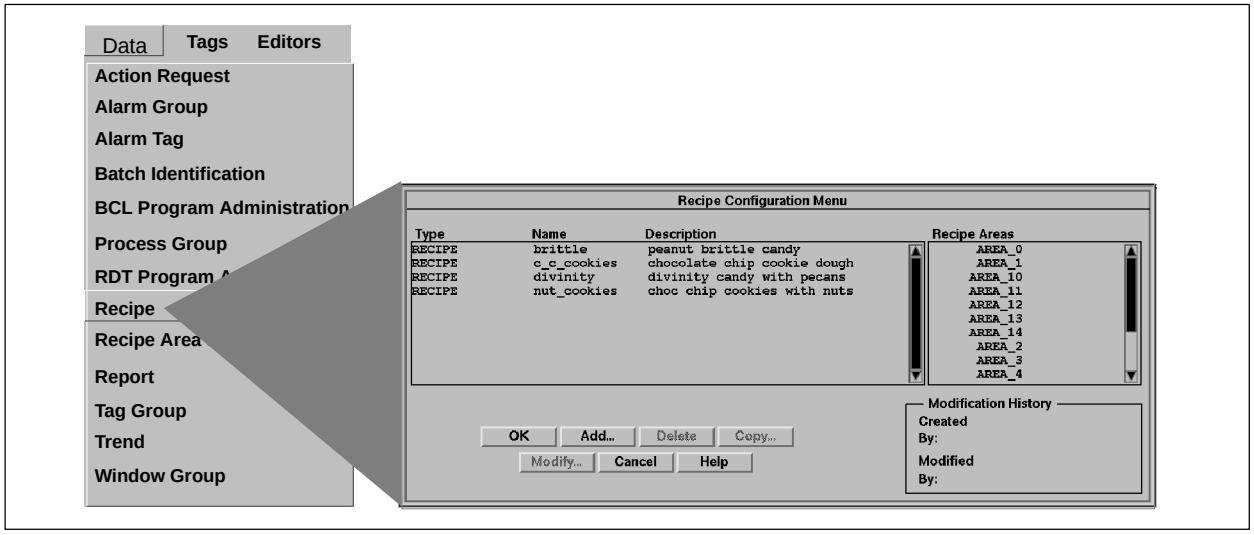


Figure 5-1 Recipe Configuration Menu Display

5.2 Using the Recipe Configuration Menu Display

Selecting an Option

Use the Recipe Configuration Menu display to perform any of the following options:

- Add new recipes and subrecipes to the system.
- Delete existing recipes and subrecipes.
- Copy components from one recipe or subrecipe to another.
- Modify existing recipes and subrecipes.

Specifying the Recipe/Subrecipe Name

To configure a recipe or subrecipe, add the name to the Recipe Configuration Menu display. A name can contain up to 12 alphanumeric characters. All characters are valid except the space character, the string -> and the following characters: , ” \ ; The name can begin with an alphanumeric character (A–Z, 0–9) or the underscore (_) character. The name can be entered in upper or lower case. Do not use the word NONE as a recipe or subrecipe name.

When you select the **Add** or **Copy** option, the system prompts you for a name. The name must be different from all other recipe, subrecipe, and recipe component names. If the name is the same as an existing recipe or subrecipe name, you receive an error message and the system rejects the entry. The entry is not case-sensitive (upper and lower case are both valid). Recipe configuration checks for embedded blanks within a recipe name and does not allow you to save a recipe with embedded blanks. This applies to any recipes that you copy, modify, or add.

Giving subrecipes and other recipe component types the same name can cause unexpected results. Always use unique names for all recipe component types.

⚠ WARNING

Subrecipes and other recipe component types must have unique names.

Using the same name for subrecipes and other recipe component types can cause unpredictable operation by the controller and the process. This could lead to serious injury or death to personnel, and/or damage to equipment.

Always use unique names for all recipe component types.

If you plan to configure the system to have the plc_rcp_req attribute of an area tag, select a recipe and download it, you must assign the recipe name using a capital letter R followed by a positive integer (for example, R4).

Specifying the Entry Type

When you add a new entry or copy the contents of an entry, you must specify the type, namely Recipe or Subrecipe. The Type field is important for the following reasons.

- You can only assign recipes to recipe areas for production; you cannot assign subrecipes to recipe areas.
- A recipe can contain a subrecipe, a subset of the recipe components. A subrecipe cannot contain a subrecipe; nesting is not valid.

The placement of the subrecipe within a list of recipe components determines which values for the common components take precedence in the production recipe. For example, if a subrecipe for a flour base comes after the recipe components for flour and water, the subrecipe values override those of the recipe.

Specifying the Recipe/Subrecipe Description

Describe every recipe and subrecipe to eliminate confusion. Valid recipe and subrecipe descriptions can contain up to 30 characters.

Adding a Recipe/Subrecipe

To add a recipe or subrecipe, follow these steps.

1. Click the **Add** button located at the bottom of the Recipe Configuration Menu display. The Add Recipe dialog box appears (Figure 5-2).
2. Enter the appropriate data in the fields provided. Field definitions for Name, Description, and Type are the same as used for the Recipe Configuration Menu display on page 5-2.

Figure 5-2 Add Recipe Dialog Box

Deleting a Recipe/Subrecipe

To delete a recipe or subrecipe from the Recipe Configuration Menu display, follow these steps:

1. Click the recipe or subrecipe you want to delete. The recipe or subrecipe you select highlights in reverse video.
2. Click the **Delete** button located at the bottom of the Recipe Configuration Menu display. The **Are You Sure You Want To Delete This?** prompt appears.

Copying a Recipe/Subrecipe

To copy a recipe or subrecipe from the Recipe Configuration Menu display, follow these steps:

1. Click the recipe or subrecipe you want to copy. The recipe or subrecipe you select highlights in reverse video.
2. Click the **Copy** button. The Copy Recipe To dialog box appears (Figure 5-3).
3. Enter name, description, and type in the appropriate fields.

⚠ CAUTION

Copying a recipe to a subrecipe or changing a subrecipe to a recipe can cause unexpected results, such as nested recipes.

Improperly copying, or improperly modifying, a recipe or subrecipe could result in unintended ingredients being added to the product.

Do not copy a recipe to a subrecipe or change a subrecipe to a recipe.

Figure 5-3 shows how to access the Copy Recipe To dialog box from the Recipe Configuration Menu display.

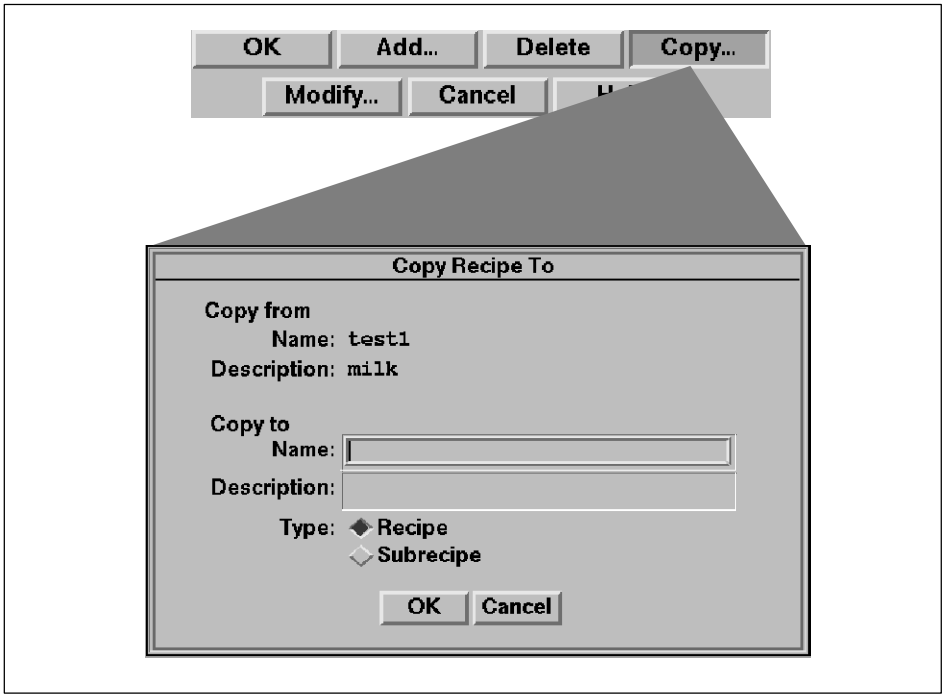


Figure 5-3 Copy Recipe to Dialog Box

Modifying a Recipe/Subrecipe

To modify an existing recipe or subrecipe from the Recipe Configuration Menu display, follow these steps:

1. Click the recipe or subrecipe you want to modify. The item you select highlights in reverse video.
2. Click the **Modify** button. The Recipe Configuration Editor display appears (Figure 5-4).
3. Enter data in the appropriate fields. Field descriptions are defined below.

Name The Name field automatically displays the recipe or subrecipe that you selected from the Recipe Configuration Menu display.

Description The Description field is a character string that describes the recipe or subrecipe. The description field display fills automatically.

Component The Component field is a character string identifying an ingredient, processing parameter, or control mechanism. The component name must match the component name in associated recipe areas. Use up to twelve characters to describe a component.

Type The Type field defines the characteristics of the component value and, when applicable, the low and high limits. Examples of types include integer, real, boolean, RCB_WAIT, RCB_BEG, RCB_END, and subrecipe.

Value The Value field identifies the initial recipe value. If this field is left blank, you must manually enter values before downloading recipes.

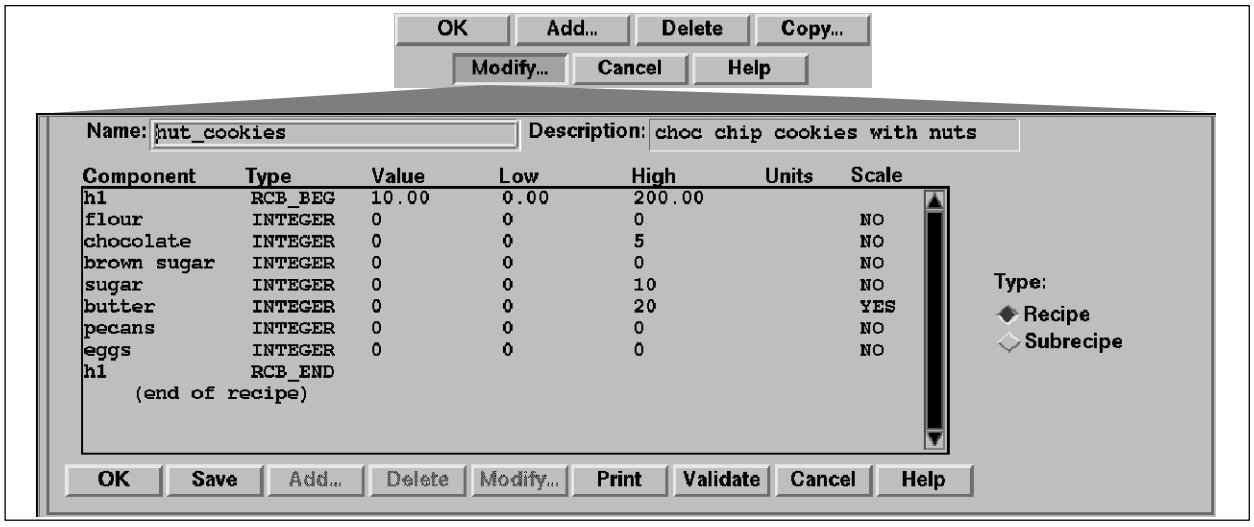


Figure 5-4 Recipe Configuration Editor Display

Low The Low field identifies the minimum value for operator entries.

High The High field identifies the maximum value for operator entries.

Units The Units field identifies the engineering units associated with the component value.

Scale The Scale field shows whether the associated integer or real component value is multiplied by the scale factor when either the production recipe is created or the scale factor is changed.

Type The **Type** radio button indicates whether the selected entry is a recipe or subrecipe.

The pushbutton selections that are located at the bottom of the Recipe Configuration Editor display are defined below.

Add The position of the components in the list is important for controlled download. You can add components anywhere in an existing list. When you select **Add**, a dialog box appears that allows you to enter data.

Delete The **Delete** option deletes the currently selected entry and all associated values.

Modify When you select **Modify**, a dialog box appears that allows you to change component values for the currently selected entry.

Print When you select **Print**, the system sends the current recipe or subrecipe definition to the printer that you assigned during printer configuration.

Validate The **Validate** option:

- Compares components in the current recipe with components in all associated recipe areas.
- Compares the component types, values in relation to range, and engineering units of the current recipe selection against those of the associated download tag and attribute in the recipe areas.
- Identifies mismatches.

NOTE: Select **Validate** only after you have completed recipe configuration and linked recipes with recipe areas on the Recipe Configuration Menu display.

5.3 Configuring Recipes

Adding Components

To add components and component values to recipes or subrecipes from the Recipe Configuration Editor display, follow these steps.

1. Click the **Add** button located at the bottom of the Recipe Configuration Editor display. The Add Component Values dialog box appears (Figure 5-5).

2. Enter data in the appropriate fields. The fields are not case-sensitive (upper and lower case are both valid). Field descriptions are explained on the following pages.

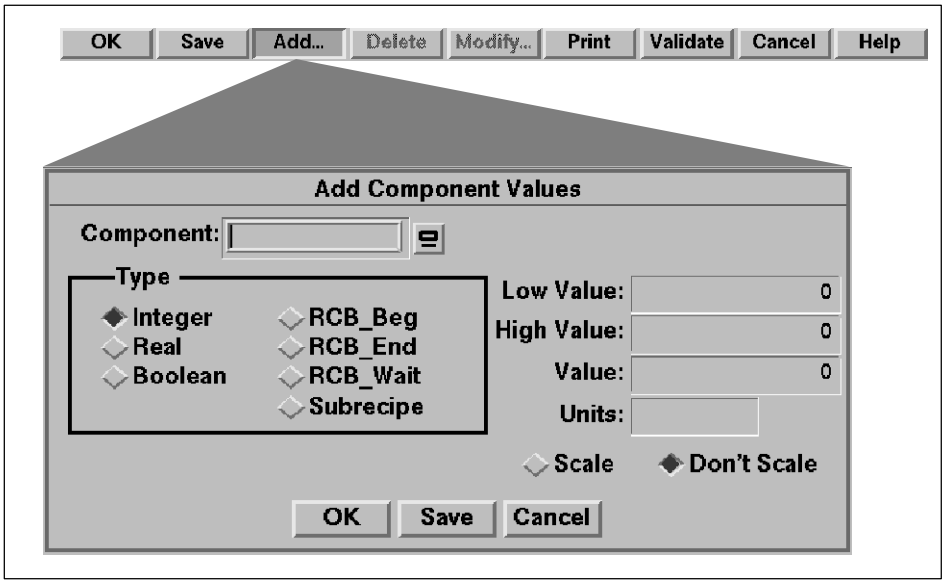


Figure 5-5 Add Component Values Dialog Box

Recipes and subrecipes consist of ingredients, such as flour and water, processing parameters, such as cook times and temperatures, and recipe controls, such as subrecipes or block control components.

Each recipe component in the list must be in at least one of the associated recipe areas. The only recipe components used are those in the area to which you download the recipe.

Select component names carefully, especially if the recipe is associated with more than one recipe area. Make sure that the download tags and attributes of matching components are associated with the same data type and engineering units. This ensures that the component values receive the same value interpretation.

Giving subrecipes and other recipe component types the same name can cause unexpected results. Always use unique names for all recipe component types.

⚠ WARNING

Subrecipes and other recipe component types must have unique names.

Using the same name for subrecipes and other recipe component types can cause unpredictable operation by the controller and the process. This could lead to serious injury or death to personnel, and/or damage to equipment.

Always use unique names for all recipe component types.

Order is important with respect to the recipe control mechanisms **RCB_BEG**, **RCB_END**, **RCB_WAIT**, and the duplicate component names in recipes and subrecipes. Enter those components in the order required for download.

NOTE: If you select **RCB_WAIT**, you must choose another data type before you enter additional components.

Choosing the Data Type

OSx has seven data types available for recipe and subrecipe components.

- Integer (16-bit signed integer)
- Real (32-bit floating point)
- Boolean (boolean; discrete)
- RCB_WAIT (discrete)
- RCB_BEG (discrete)
- RCB_END (discrete)
- Subrecipe (subrecipe)

The data type of a recipe component must match the data type of the tag attribute in the OSx database, regardless of the data type of the tag attribute in the controller. This rule applies for all attributes of all tag types. It allows for flexibility in scaling between the controller representation of the data value and the format of the value presented to the user.

For example, the database stores timer current and preset values as FLOAT32, although the controller may store them as integers. The integer value in the controller can represent 1 millisecond, 100 milliseconds, 1 second, or some other time interval, depending on the controller type and program. Representing these quantities as floating point values in the database enables the system to display them in a format that the operator can understand.

Be sure that the data type that you select for a recipe component matches the data type of the tag attribute in the OSx database. Remember that this data type may be different from the data type in the controller.

Entering Range Limits

If you make a manual entry at recipe activation, that entry must come within the range limits defined here. Range limits are used to check manually entered values. If the scale flag is set, these values are scaled before the manual entry is checked.

The scale factor is a floating point value. If the scale flag is set to **YES**, the component values are multiplied by the scale factor. Scaled integer values are rounded to the nearest whole number. Scaled integer values of 0.5 or greater are rounded up; scaled integer values of less than 0.5 are rounded down. Scaled real values are rounded to the nearest hundredth.

Entering the Component Value

For each component, enter the value that you want to download when you produce the recipe.

NOTE: Do not enter a component value outside the original low and high ranges of the tag linked to this component. If you enter a value outside the original low and high ranges the recipe download fails. A pop-up warning appears to alert you that the component value is out of range.

Entering Units

All components have associated engineering units except boolean and discrete components.

Setting the Scale Flag

Use the **Scale Flag** to indicate which real or integer component values must be multiplied by the scale factor (configured and defined with area tags) when the production recipe is created. Scaling is also performed on the associated range limits before any manual entries are checked against them.

Modifying Component Values

To modify components and component values of recipes or subrecipes from the Recipe Configuration Editor display, follow these steps.

1. Click the recipe or subrecipe you want to modify. The item you select highlights in reverse video.
2. Click the **Modify** button located at the bottom of the Recipe Configuration Editor display. The Modify Component Values dialog box appears (Figure 5-6).
3. Enter data in the appropriate fields. Field definitions for **Modify** and **Add** are the same as described on page 5-9.

NOTE: When you configure a tag for a recipe, do not configure it for use in an action request.

Figure 5-6 shows how to access the Modify Configuration Values dialog box from the Recipe Configuration Editor.

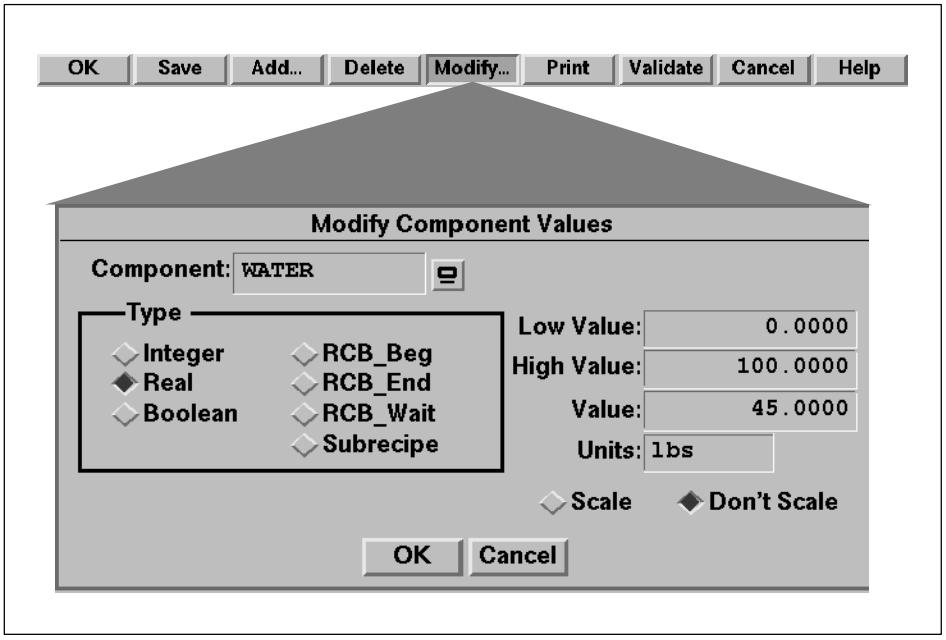


Figure 5-6 Modify Component Values Dialog Box

Linking Recipes and Recipe Areas

Before validation, you must link recipes with recipe areas that are used to produce the recipes. You do this from the Recipe Configuration Menu display after completing both the recipe area configuration and recipe/subrecipe configuration.

To link recipes with recipe areas, follow these steps.

1. Select **Data->Recipe** from the menu bar. The Recipe Configuration Menu display appears (Figure 5-7).
2. Click the recipe you want to link to a recipe area. The recipe you select highlights in reverse video.
3. Click the recipe area(s) you want to link with the selected recipe. The linked recipe area shows in yellow.

When you link and validate a recipe and recipe area, the recipe area shows in white. Chapter 6 describes recipe validation.

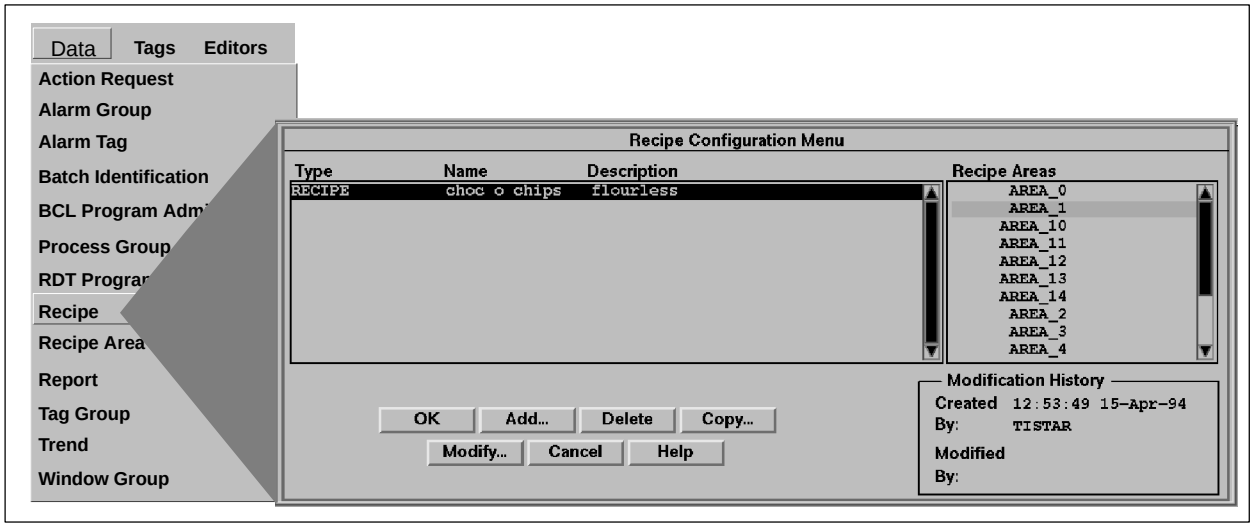


Figure 5-7 Linking Recipes and Recipe Areas

Chapter 6

Validating Recipes

6.1 Recipe Validation Overview 6-2
Purpose 6-2
Preparing to Validate Recipes 6-2

6.2 Validating Recipe Components 6-3
Creating a Production Recipe 6-3
Evaluating Validation Results 6-6

6.1 Recipe Validation Overview

Purpose

When you select the **Validate** option, the system compares the components of the current recipe with the components of all linked recipe areas. The purpose of validation is to help ensure compatibility between the recipe and any associated recipe area.

Preparing to Validate Recipes

You can validate recipes in either the Offline or Operate state. Before you can validate a recipe, you must configure both the recipe and the recipe area, and you must link the recipe with the recipe area. If necessary, you can modify validated recipes.

6.2 Validating Recipe Components

Creating a Production Recipe

To validate recipe/recipe area combinations, use the **Validate** option. This simulates creating production recipes for each linked recipe area. A production recipe contains all the recipe component values that download when the area is activated. Recipes and recipe areas merge to create a production recipe.

While validating the production recipe, the system compares the components of the recipe with those of the recipe area. The system verifies that all components required to produce the designated recipe are present in the linked areas.

NOTE: If the same recipe component appears more than once in a recipe, it is listed only once in the Recipe Validation display. The value displayed beside the component is the last value downloaded.

Validating Recipe Components (continued)

To validate a recipe, follow these steps.

1. Select **Data->Recipe** from the menu bar. The Recipe Configuration Menu display appears.

2. Be sure that you have linked the recipe with one or more recipe areas as described on page 5-15.

3. Click the **Modify** button located at the bottom of the Recipe Configuration Menu display. The Recipe Configuration Editor display appears.

4. Click the **Validate** button. The Recipe Validation display appears (Figure 6-1).

Field descriptions are defined on the following page.

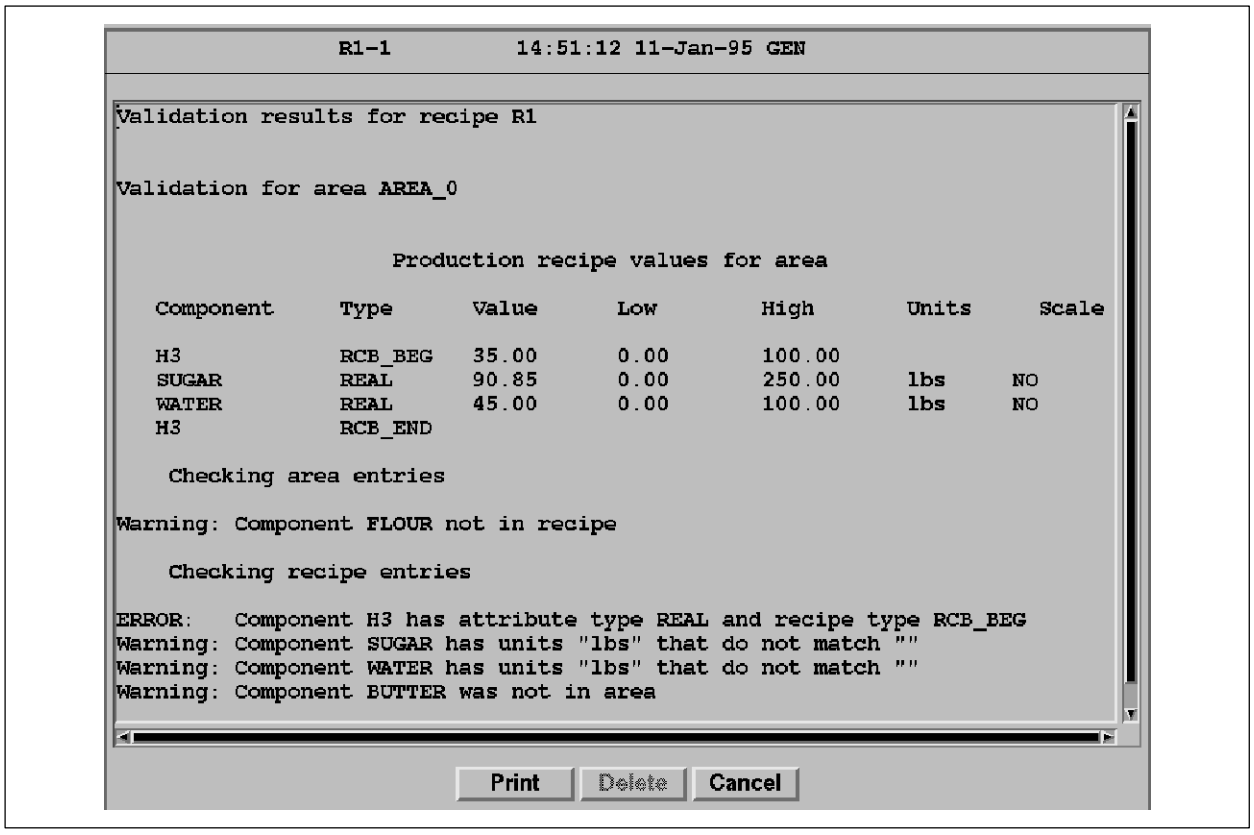

R1-1 14:51:12 11-Jan-95 GEN
Validation results for recipe R1
Validation for area AREA_0
Production recipe values for area
Component Type Value Low High Units Scale
H3 RCB_BEG 35.00 0.00 100.00
SUGAR REAL 90.85 0.00 250.00 lbs NO
WATER REAL 45.00 0.00 100.00 lbs NO
H3 RCB_END
Checking area entries
Warning: Component FLOUR not in recipe
Checking recipe entries
ERROR: Component H3 has attribute type REAL and recipe type RCB_BEG
Warning: Component SUGAR has units "lbs" that do not match ""
Warning: Component WATER has units "lbs" that do not match ""
Warning: Component BUTTER was not in area
Print
Delete
Cancel

Figure 6-1 Recipe Validation Display

Recipe Name Date/Time Recipe name appended with a unique identifier (R1–1 shown in Figure 6-1); the first version of a recipe validation held by the system uses an identifier of 1; on systems with multiple operator stations, it is possible to have several versions of the same recipe validation report in the system simultaneously.

Warning: Production Recipe Conflicts A warning appears if the recipe being validated was edited and its old values are currently used in a production recipe or the recipe component.

Another warning appears if an area has the old version of this recipe in its production recipe.

Production Recipe Values for Area This field displays the final simulated production values after the validation and download process completes. The values fill automatically.

Checking Area Entries This field displays a list of all components in the designated recipe area not used in the recipe.

Checking Recipe Entries Table 6-1 shows a list of possible problems.

Table 6-1 Examples of Possible Problems

Warning/Error Description	Warning/Error Type	Recipe Download
Components in a designated recipe that are not in the recipe area	Warning	Components are omitted at Recipe Select/Load and the remainder of the recipe successfully downloads
Recipe component high range is higher than component tag high range	Warning	Allows successful download
Mismatches between data types (integer vs. real)	Error	Validation fails; recipe cannot be downloaded
Mismatches between units	Warning	Allows successful download

Evaluating Validation Results

The Recipe Validation display (Figure 6-1) shows production recipe values for the associated recipe area and shows inconsistencies between the components in the recipe and the associated recipe area.

Validation evaluates inconsistencies before permitting a recipe to be used. Recipe entry checks produce one of three outcomes:

- **No error** The recipe component configured correctly.
- **Warning** The recipe component does not match the area component, but the problem does not necessarily make the recipe invalid.
- **Error** The recipe component creates a fatal problem, and the recipe cannot be validated until the problem is resolved.

The following errors cause a recipe not to validate.

- Data type mismatch: for instance, an integer type component cannot be validated against a real type attribute.
- A non-existent subrecipe is listed as a component.
- Recipe control block nesting or mismatch.

Not all components of the recipe area are required for producing the recipe. Some extra components show in the Checking Area Values list section of the screen. Similarly, components can exist in the recipe but not in the recipe area. These components are not used in the production recipe; they show up in the Checking Recipe Components list.

⚠ WARNING

Mismatches listed on the validation screen do not necessarily make the recipe/recipe area combination invalid. Nevertheless, these mismatches could cause unpredictable controller operation, which could result in death or serious injury to personnel, and/or damage to equipment.

Evaluate the validation screen listings and determine whether modifications are required for producing the production recipe with the intended result.

Ensure that recipes validate properly and that component mismatches are identified and resolved before using the components in a production recipe.

Chapter 7

Selecting and Loading Recipes

7.1 Selecting Recipes for Download 7-2
Preparing to Download Recipes 7-2
Accessing the Recipe Selection Display 7-2
Preplanning a Recipe Download 7-4
Clearing the Selected Recipe 7-5
Loading a Recipe 7-6

7.2 Loading Recipes: Troubleshooting 7-8
Loading Recipes: Bad Scale Factor 7-8
Loading Recipes: No Production Recipe Display 7-10
Loading Recipes: Delay for Role Change 7-10

7.3 Manually Loading Recipes for Production 7-11
Overview of the Download Process 7-11
Modifying Production Recipe Values 7-11
Modifying Scale Values 7-12
Loading a Production Recipe 7-13
Printing a Production Recipe 7-13
Canceling a Download Request 7-13
Suspending a Recipe Download 7-13
Continuing a Recipe Download 7-13

7.4 Loading Recipes by Controller Action 7-14
Overview of the Download Process 7-14
Using the Controller to Select and Download Recipes 7-14
Using the Controller to Cancel a Download Request 7-14
Using the Controller to Download Recipes Already Selected 7-15
Using the Controller to Modify Recipe Values 7-15

7.5 Acknowledging Recipe Alarm Tags 7-16

7.1 Selecting Recipes for Download

Preparing to Download Recipes

Before you download recipes, you must validate the recipe and ensure that the system is in the Operate state.

Accessing the Recipe Selection Display

The Recipe Selection display allows you to specify the next recipe you want to download to the designated recipe area. You can select only one recipe from a recipe area at a time.

To access the Recipe Selection display, follow these steps.

1. Set the system state to Operate. OSx displays the navigation area.
2. Select the **Directory** icon from the navigation area. A directory of OSx display types appears (Figure 7-1).
3. Select the **Recipe** option. A list of area tags appears (Figure 7-1).
4. Select the appropriate area tag. The Recipe Selection display appears (Figure 7-2).

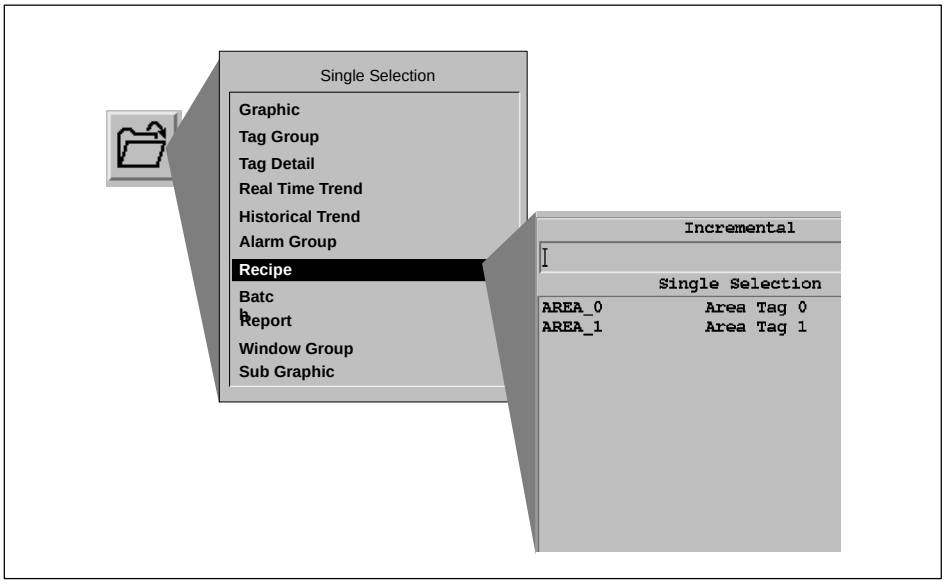


Figure 7-1 Recipe Directory

Figure 7-2 shows the Recipe Selection display.

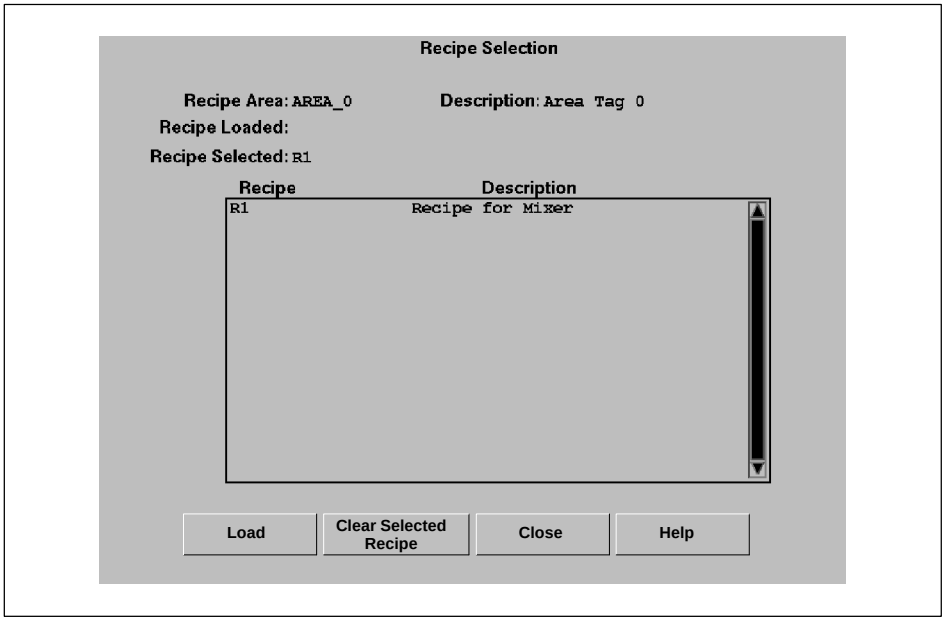


Figure 7-2 Recipe Selection Display

Recipe Area The Recipe Area field displays the name of the designated recipe area; the field automatically fills when the screen is accessed.

Description The Description field displays a description of the recipe area; the field automatically fills when you access the screen.

Recipe Loaded The Recipe Loaded field displays the name of the recipe previously downloaded.

Recipe Selected The Recipe Selected field displays the recipe currently selected.

Recipe The Recipe field lists all recipes validated for that area.

Description The Description field displays the description of the recipe as entered in the Recipe Configuration Directory.

Load The **Load** button creates a production recipe and accesses the Production Recipe Preparation display.

Clear Selected Recipe The **Clear Selected Recipe** button allows the operator to deselect the recipe for the recipe area.

Planning a Recipe Download

The Recipe Selection display allows you either to plan recipe downloads in advance or to select and immediately download a recipe to the designated recipe area.

To plan a recipe download in advance, follow these steps.

1. Set the system state to Operate. OSx displays the navigation area.
2. Select the **Directory** icon from the navigation area. A directory of OSx display types appears (Figure 7-3).
3. Select the **Recipe** option. A list of area tags appears.
4. Select the area tag that you want to download. The Recipe Selection display appears (Figure 7-4).
5. Select the recipe that you want to download to the selected recipe area. OSx links the recipe to the recipe area.
6. To exit the Recipe Selection display, click the **Directory** icon from the navigation area and select any item listed.

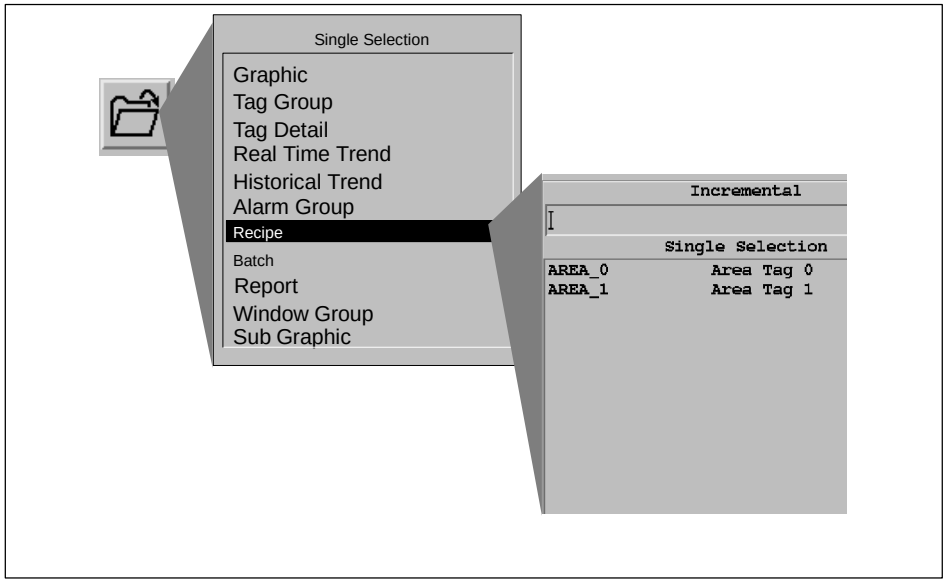


Figure 7-3 Selecting a Recipe for Download

OSx creates the production recipe but does not access the Production Recipe Preparation display.

When you are ready to load the recipe, follow these steps.

1. Set the system state to Operate. OSx displays the navigation area.
2. Select the **Directory** icon from the navigation area.A directory of OSx display types appears (Figure 7-3).
3. Select the **Recipe** option. A list of area tags appears.
4. Select the area tag that you want to download. OSx accesses the Production Recipe Preparation display. Your controller can also make a download-only request for the area (downloading values from the pre-selected recipe) by setting a value of –3 to the area's **plc_rcp_req** attribute.

Clearing the Selected Recipe

The **Clear Selected Recipe** pushbutton on the Recipe Selection Display (Figure 7-4) allows you to cancel the current recipe selection. The **Clear Selected Recipe** option deselects the recipe for the recipe area.

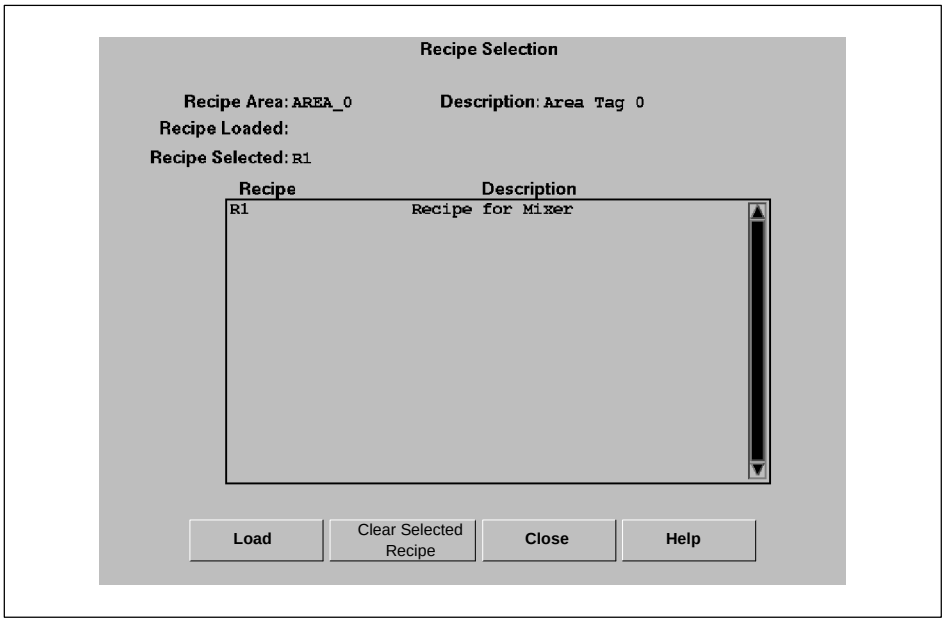


Figure 7-4 Downloading a Recipe

Loading a Recipe

To select and immediately download a recipe, follow these steps.

1. Set the system state to Operate. OSx displays the navigation area.
2. Select the **Directory** icon from the navigation area. A directory of OSx display types appears.
3. Select the **Recipe** option. A list of area tags appears.
4. Select the area tag that you want to download. The Recipe Selection display appears.

   If a production recipe already exists, the Production Recipe Preparation display appears, and you can skip steps 5 and 6.
5. Select the recipe you want to download. The recipe is linked to the recipe area.
6. Select the **Load** pushbutton. OSx creates the production recipe and accesses the Production Recipe Preparation display (Figure 7-5).

Field descriptions for the Production Recipe Preparation display are defined below.

Recipe Area The Recipe Area field displays the name of the designated recipe area; the field automatically fills when the screen is accessed.

Description This Description field identifies the recipe area; the field automatically fills when the screen is accessed.

Recipe The Recipe field displays the recipe you selected for download from the Recipe Selection display; the field automatically fills when the screen is accessed.

Description This Description field identifies the recipe selected; the field automatically fills when the screen is accessed.

Scale The Scale field displays the scale factor used to adjust values for the production recipe. The scale factor can be changed manually before download. If a scale factor attribute of the recipe area tag used is networked, the scale factor can also be adjusted by the controller. See page 7-8. Only those components configured with scaling are adjusted. Scaled integer values of 0.5 or greater are rounded up to nearest whole number and those less than 0.5 are rounded down; scaled real values are rounded to the nearest hundredth.

The **Select Recipe** pushbutton allows access to the Recipe Selection display if the production recipe is not being downloaded.

The **Modify Value** pushbutton allows you to change the value of a selected component.

The **Modify Scale** pushbutton allows you to change the scale value.

The **Start Download** pushbutton checks the production recipe for null values indicating that an operator entry is required, begins the download to the controller locations, and inserts the production recipe values into the download tags for the recipe area.

The **Stop Request** pushbutton terminates the currently active download. The **Stop Request** selection can only terminate on a **RCB_BEG** block that is waiting for the area to become available.

The **Hold** pushbutton suspends a recipe download indefinitely (until continue is requested). You can suspend a recipe either before the load is requested or on a **RCB_BEG** block that is waiting for the area to become available. If you request a hold before the download is started, the download is suspended when the next **RCB_BEG** block is encountered.

The **Continue** pushbutton removes the area from the hold state.

The **Print** pushbutton displays the Print dialog box, allowing you to send the production recipe to the selected destination.

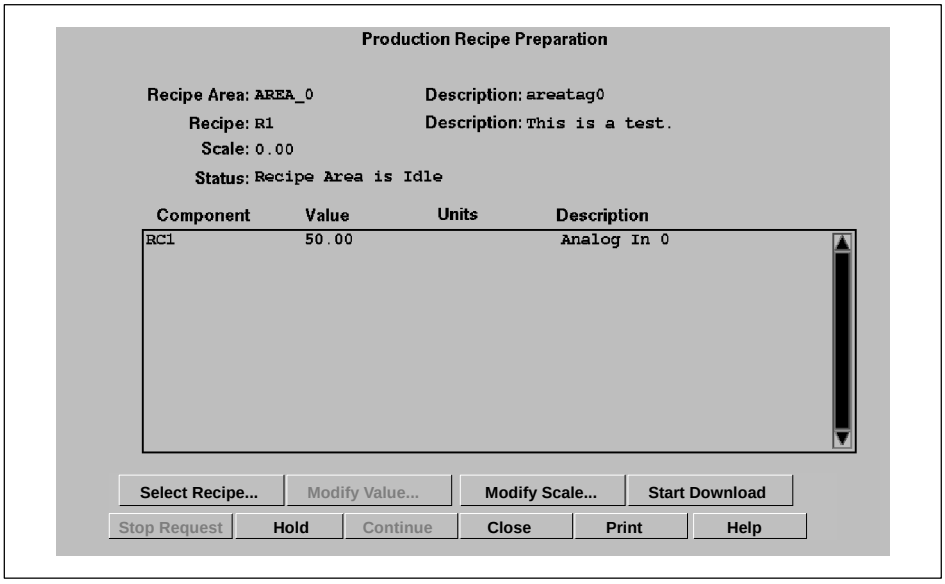


Figure 7-5 Production Recipe Preparation Display

7.2 Loading Recipes: Troubleshooting

Loading Recipes: Bad Scale Factor

Recipe scale factors are expressed in floating point numbers, which range from -1×10^{38} to 1×10^{38}. Any number outside this range is Not a Number (NaN).

Scale factors are specified for recipe components by these methods.

- The operator enters a scale factor before loading a recipe.
- The plant engineer specifies a scale factor when configuring the recipe from the OSx recipe dialog boxes.
- The value is uploaded from a controller.
- The plant engineer specifies a scale factor when creating a program of one of these types: report or custom program (e.g., C-language, BCL, or RDT).

If an operator accesses the Production Recipe Preparation dialog box and `NaN` appears in the Scale field (Figure 7-6), the scaling factor is invalid.

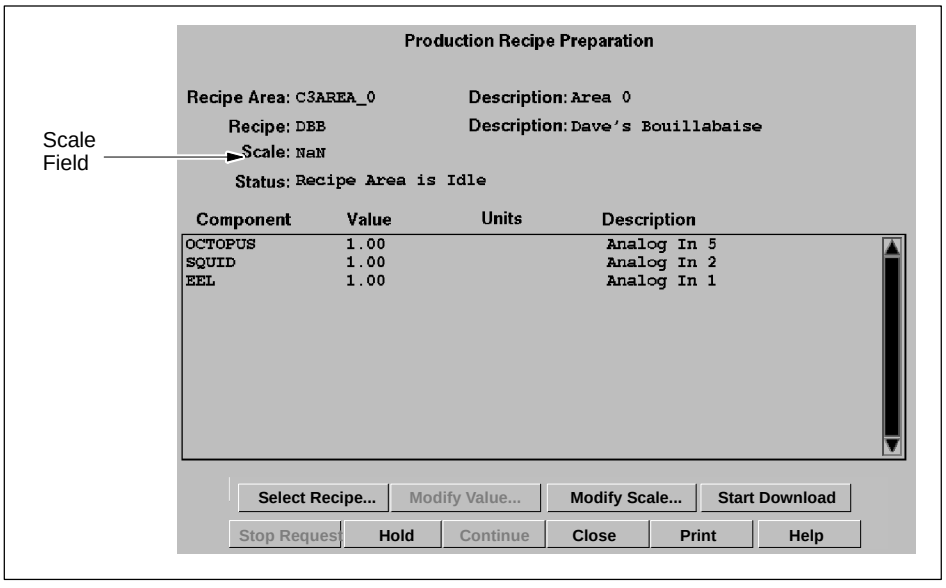


Figure 7-6 Recipe Scale Field Contains NaN

If the operator attempts to load the recipe, the system cancels the recipe download and displays the following message:

Invalid scale factor encountered — no scaling done.

If the system attempts an automatic recipe download due to a BCL request, for example, and an invalid scale factor is in the database, the system cancels the recipe download, sets a recipe system alarm, and displays the following message:

Invalid scale factor for recipe area <recipe_area_name> ; download aborted.

If the operator has entered the invalid scale factor, or if the invalid value was entered during recipe configuration, the operator can enter a valid factor in the Modify Scale dialog box (Figure 7-7).

If the invalid scale factor comes from one of the other sources listed above, the operator can enter a new value, but it may be overwritten by an automated mechanism before the valid value can be used. You need to identify the origin of the invalid scale factor, such as a BCL program, the database of a remote MIS system (RDT), or a controller program, and make your corrections there.

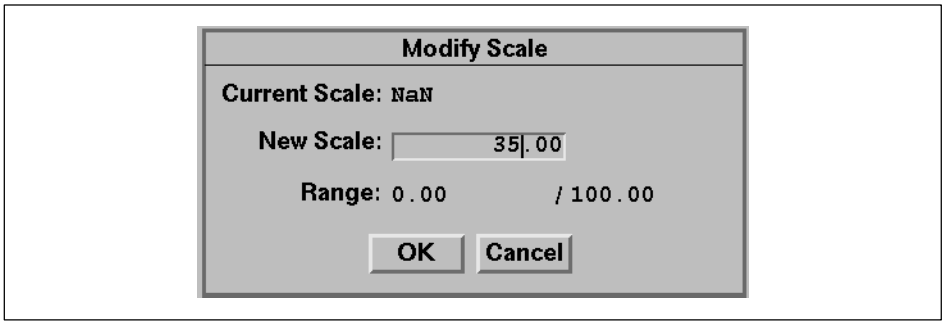


Figure 7-7 Modify Scale Dialog Box for Recipes

Loading Recipes: Troubleshooting (continued)

Loading Recipes: No Production Recipe Display

If you select the **Load** button on the Recipe Selection dialog box and the Production Recipe Preparation dialog box does not appear, the system could be having difficulty creating the production recipe for the area. This can occur if you change the recipe or area while the system is displaying the Recipe Selection box.

You can revalidate the recipe for the area to prevent a recurrence. However, the system may not display the area on other stations, and the controller may not be able to use the area until you either transition the system from the Operate state to the Offline state and back to Operate again, or perform the following procedure.

1. At the station where the problem originally occurred, display the area again. OSx displays the Production Recipe Preparation dialog box. No components appear since the production recipe has not been created.

2. Close the Production Recipe Preparation dialog box by displaying another screen, such as a graphic or tag detail. The area is now available for use on any station or by the controller.

Loading Recipes: Delay for Role Change

If you select **Start Download** from the Production Recipe Preparation dialog box at the same time that a user on another station confirms a role change, the system does not download the recipe. The Production Recipe Preparation dialog box does not display the Start Download messages. The system writes Error 11254, which you can ignore, to the **log.out** file. Wait for the role change to complete and then try to download the recipe again.

7.3 Manually Loading Recipes for Production

Overview of the Download Process

When you select the **Start Download** pushbutton from the Production Recipe Preparation display, the system checks the production recipe for null values. Null values must be replaced by operator-entered values. All operator entries are recorded on the Operator Change Log.

After the production recipe is checked and all entries have values, the system initiates the download to the memory locations defined for the associated recipe area. If the production recipe contains the recipe control blocks, **RCB_WAIT**, **RCB_BEGIN**, and **RCB_END**, the control blocks determine the order in which recipe components are downloaded. The system downloads all recipe components listed in front of an **RCB_WAIT** and between an **RCB_BEGIN** and **RCB_END** before continuing the download. Except for the **RCB_WAIT**, the operator receives messages for all waits during the download.

NOTE: If you select **Start Download** from the Production Recipe Preparation dialog box at the same time that a user on another station confirms a role change, the system does not download the recipe. The Production Recipe Preparation dialog box does not display the **Start Download** messages. The system writes **Error 11254**, which you can ignore, to the **log.out** file. Wait for the role change to complete and then try to download the recipe again.

Modifying Production Recipe Values

To change component values before downloading a production recipe, follow these steps.

1. Click the component you want to modify from the Production Recipe Preparation display. The component you selected highlights in reverse video.
2. Click the **Modify Value** pushbutton. The Modify Production Value dialog box appears (Figure 7-8).
3. Enter a new value in the appropriate field. The values modified show on the Production Recipe Preparation display.

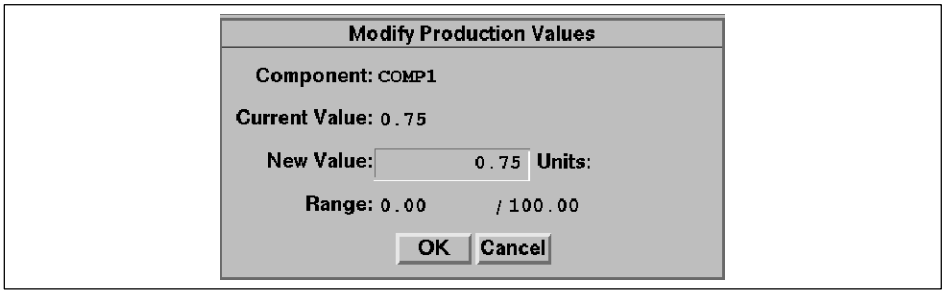


Figure 7-8 Modify Production Value Dialog Box

Modifying Scale Values

To change scale values before you download a production recipe, follow these steps.

1. Click the **Modify Scale** pushbutton. The Modify Scale dialog box appears (Figure 7-9).

2. Enter a new scale value in the appropriate field.

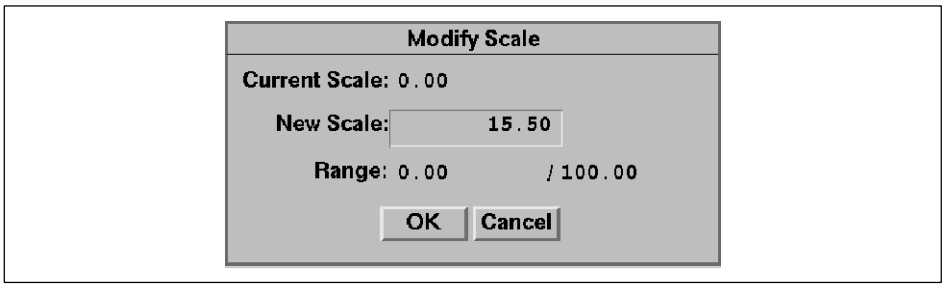


Figure 7-9 Modify Scale Dialog Box

NOTE: When you select **OK** on the Modify Scale dialog box, OSx applies the scale factor against the original recipe value that was entered in the recipe configuration, not against the current value.

If you modify an individual scalable value from the download screen after OSx applies the scale factor, it is not rescaled.

If you modify an individual scalable value from the download screen before OSx applies the scale factor, it is rescaled from its original value. The rescaled value is the value that is downloaded.

Loading a Production Recipe

To download the production recipe values after all operator entries and modifications are made, select the **Start Download** pushbutton from the Production Recipe Preparation display.

Printing a Production Recipe

To print the production recipe shown on the Production Recipe Preparation display, select the **Print** pushbutton.

Canceling a Download Request

To cancel a download that is currently in progress, select the **Stop Request** pushbutton from the Production Recipe Preparation display. The recipe download can only be canceled on a **RCB_BEG** block. If a stop request command is initiated during an active download, the command is not recognized until the system evaluates a **RCB_BEG** block and is waiting for the In_use bit to become available.

Suspending a Recipe Download

To suspend a download, select the **Hold** pushbutton from the Production Recipe Preparation display. The recipe download can only be suspended on a **RCB_BEG** block. If a hold is requested before the download has started, the download is suspended when the next **RCB_BEG** block is encountered. If a hold is requested during an active download, the hold request is not recognized until the system evaluates a **RCB_BEG** block and waits for the In_use bit to become available.

During a hold state, the timeout for the **RCB_BEG** is also suspended. The time remaining until timeout is stored. A Continue request cancels the hold state and allows the time remaining to proceed until timeout.

Continuing a Recipe Download

To cancel a hold request (to continue a suspended recipe download), select the **Continue** pushbutton from the Production Recipe Preparation display.

NOTE: If the controller changes the hold status of a recipe area while the Production Recipe Preparation dialog box is displayed, then the **Hold** and **Continue** buttons are not updated correctly. There is no indication on the screen that this has happened, except that the buttons do not work as expected.

To correct the situation, redisplay the Production Recipe Preparation screen. This causes the **Hold** and **Continue** buttons to display correctly.

7.4 Loading Recipes by Controller Action

Overview of the Download Process

A controller can use a networked location to request a recipe download (either selecting the recipe to download, or downloading a recipe that has already been selected) or to abort a controller-initiated download.

To have a controller control a recipe area, you must define the plc_rcp_req attribute during area tag configuration.

Using the Controller to Select and Download Recipes

You can select and download a recipe by writing a value to the plc_rcp_req location. The value in the plc_rcp_req tag must be a positive integer, and there must be a recipe identified with an "R" plus the integer (such as R1 or R2).

When the controller writes an integer (for example, 1) to plc_rcp_req, OSx selects and downloads the requested recipe (for example, R1). The recipe area must not be "owned" by another device.

Three conditions generate errors when downloading recipes in this manner.

- Another device (operator station or controller) owns the recipe area.
- The recipe requested (such as R1) does not exist.
- The recipe requested has empty values. (All recipe values must have a value assigned before the recipe downloads.)

NOTE: If you modify individual recipe values for a recipe on the Production Recipe Preparation display, you can send these modified values to the controller only by selecting the **Start Download** pushbutton on the Production Recipe Preparation display. A controller-initiated download request can only download the original recipe values of a selected recipe.

Using the Controller to Cancel a Download Request

The controller can cancel a controller-generated download request by writing a –1 to the plc_rcp_req location. The controller can only cancel download requests that it has initiated. The recipe download can only be canceled on a **RCB_BEG** block. If a cancel is requested during an active download, the cancel request is not recognized until the system evaluates a **RCB_BEG** block and waits for the In_use bit to become available.

Using the Controller to Download Recipes Already Selected

If a recipe has already been selected, you can download a recipe by writing a –3 to the plc_rcp_req location. The download-only request does not require that you name recipes with an "R" (for example, R1), as when you select and download.

When the controller writes a –3 to plc_rcp_req, OSx downloads the recipe that had been previously selected. The recipe area must not be "owned" by another device.

Three conditions can cause errors when the controller downloads selected recipes.

- Another node or controller owns the recipe area.
- No recipe has been selected.
- The selected recipe has empty values. (All recipe values must have a value assigned before the recipe can be downloaded.)

NOTE: If you modify individual recipe values for a recipe on the Production Recipe Preparation display, you can send these modified values to the controller only by selecting the **Start Download** pushbutton on the Production Recipe Preparation display. A controller-initiated download request can only download the original recipe values of a selected recipe.

Using the Controller to Modify Recipe Values

You can use your controller to control the scaling of recipe values prior to creating a production recipe. (You must have created a scale_factor attribute from the recipe area tag). Your controller updates the scale factor attribute of this tag. For example, a factor of 1.0 maintains the original values, and a factor of 2.0 doubles all recipe values that can be modified.

The scale_factor value is used when you create a production recipe (either by selecting **Load** from the Recipe Selection display, or by the controller making a valid load request). Once a production recipe has been created, re-scaling the recipe can only be performed from the Production Recipe Preparation display, or by changing the scale factor in the controller between download requests of an already existing production recipe.

7.5 Acknowledging Recipe Alarm Tags

If a tag configured for alarming has a non-networked status attribute and the tag goes into alarm, the system does not clear the Alrm_Unack (alarm unacknowledged) bit of the status word under the following conditions:

- The operator does not acknowledge the alarm, and
- The system transitions from the Operate state to the Offline state and then back to the Operate state.

The Alrm_Unack bit is the 0x2 bit of the tag's status attribute. The system sets this bit to one when the alarm has not been acknowledged. For networked status attributes, the system sets this bit to zero if the alarm is not acknowledged before the system state changes from Operate to Offline and then back to Operate.

Typically, tags configured for alarming have networked status attributes, and this situation does not occur. However, the status attribute for recipe area tags cannot be networked. For example, a recipe area tag goes into alarm, and the operator does not acknowledge the alarm before the system state transitions from Operate to Offline and then back to Operate again. If an operator displays a graphic that contains the recipe area tag's status attribute, the Alrm_Unack bit for the attribute is on, even though there are no current alarms to acknowledge for the recipe area tag.

When this situation occurs, you can ignore the setting of the Alrm_Unack bit.

Appendix A

Recipe Area Table

A.1 Area Attributes A-2

A.2 Bit Locations for Area Attributes A-3

A.1 Area Attributes

The area database table contains recipe area configuration information and data values. These values define the current state of each recipe area; for example, selected recipe, last loaded recipe, recipe area owner, and control states in the recipe download process. The attributes for area tags are shown in Table A-1.

Table A-1 Area Tag Attributes

Attribute	Domain	Definition
scale_factor	FLOAT32	Network value used to scale recipe values in the controller
plc_rcp_req	SINT16	Networked value used for control node control of recipe download requests and aborts
hold_req	SINT16	Hold/Continue request (0 x 8000=hold; 0=continue)
status	{Active, Alm_UnAck, Man_Set Hold_Req, Dl_Start, Hnd_Shk, Hold, Err_Comp, Blk_Miss, Req_Abt, Err_Abt, Timeout, Sec_Abt, Spare_1 Spare_2, Spare_3}	Current state of recipe area activity
ctl_status	{Active, Alm_UnAck, Man_Set, Hold_Req, Dl_Start, Hnd_Shk, Hold, Err_Comp, Blk_Miss, Req_Abt, Err_Abt, Timeout, Sec_Abt, Spare_1, Spare_2, Spare_3}	Indicates current activity of the recipe area; communication attribute used between recipe control and download process
scale_high	FLOAT32	High limit for scaled values
scale_low	FLOAT32	Low limit for scaled values
sel_rcp_ptag	SINT16	Identifies recipe currently selected for download to recipe area
act_rcp_ptag	SINT16	Identifies last recipe downloaded to recipe area
recipe_name	12 Characters	Name of recipe currently used in the production recipe
description	30 Characters	Describes the recipe used in the production recipe
area_owner	SINT16	Value indicating the device that can modify recipe values, download a recipe, or cancel a download to recipe area: node, controller, no_owner
term_num	SINT16	Value indicating the terminal of a node that controls a recipe area: number matches the terminal number, or zero for no terminal.
download_cmd	BIT16	Value indicating command sent to recipe area: download, abort, no_cmd

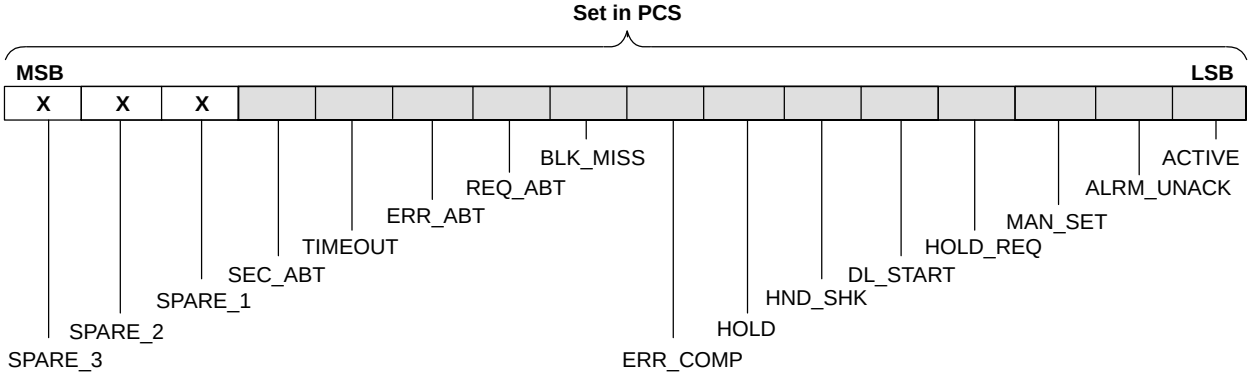
Table A-1 Area Tag Attributes (continued)

Attribute	Domain	Definition
alt_stat	UINT32	Name and color code for current highest priority alarm
component	STRING[12]	Production recipe component currently associated with the STATUS attribute
tag	CISTRING[12]	Tag name
pseudo_tag	SINT16	Tag reference ID
batch_id	CISTRING[16]	Batch ID associated with this tag

A.2 Bit Locations for Area Attributes

The bit locations for the status attribute are as follows:

Bit Name	Meaning	Bit Location (Hex)
Spare_3	(<Spare>)	0x8000
Spare_2	(<Spare>)	0x4000
Spare_1	(<Spare>)	0x2000
Sec_Abt	(Secondary Abort)	0x1000
Timeout	(Timeout)	0x0800
Err_Abt	(Error Abort)	0x0400
Req_Abt	(Requested Abort)	0x0200
Blk_Miss	(Block Mismatch)	0x0100
Err_Comp	(Completed)	0x0080
Hold	(Hold Request)	0x0040
Hnd_Shk	(Handshake)	0x0020
DL_Start	(Download)	0x0010
Hold_Req	(Suspend Recipe Download)	0x0008
Man_Set	(Manually Set)	0x0004
Alrm_UnAck	(Alarm Not Acknowledged)	0x0002
Active	(Status of Network Communication)	0x0001



Index

A

Accessing
  recipe configuration menu, 5-2
  recipe selection list, 7-2

Acknowledging alarms, recipes, 7-16

Adding
  components to recipe, 5-10
  recipe/subrecipe, 5-5

Area tag (AREA), 1-9
  attributes, 2-4, A-2
  defining, 2-3
  deleting, 2-5

B

Bit, recipe, handshake, 4-6

C

Canceling
  download request for recipe, 7-13
  recipe, 7-5

Changing
  recipe area components, 3-6
  recipe component values, 7-11
  recipe scale values, 7-12
  recipe/subrecipe, 5-8

Configuring
  recipe area attributes, A-2
  recipe area tags, 2-2
  recipe areas, 3-2, 3-4

Control block, recipe, 4-4
  nesting, 4-6
  RCB_BEG, 4-5
  RCB_END, 4-5
  RCB_WAIT, 4-4

Copying, recipe/subrecipe, 5-6

Creating
  recipe, 1-8
  recipe/subrecipe, 4-2, 5-4

D

Data type, recipe component, 5-12

Defining, recipe area tag attributes, 2-4

Deleting
  recipe area tag, 2-5
  recipe/subrecipe, 5-6

Directory, recipe, 7-2

Downloading, recipe, 4-4, 7-4, 7-6
  by plc_rcp_req attribute, 2-4, 7-14
  manually, 7-11
  phased, 4-10

E

Error, recipe, 6-6
  example, 6-5

H

Handshake, recipe, 4-6

I

Install.tag file, recipe configuration, 1-8

L

Linking, recipes with recipe areas, 5-15

Loading, recipe, 7-6

N

Nesting, recipe control blocks, 4-6

O

Operator change log, recipe, 7-11

P

Printing, recipe, 7-13

R

Recipe
- area, 1-10
  - configuration, 3-2, 3-4
  - status, A-2
- area tag (AREA), 1-9, A-2
- communication handshake, 4-6
- component, 1-4
  - adding, 3-4, 5-11
  - data type, 5-12
  - download attribute, 3-4
  - download tag, 3-4
  - modifying, 3-6
  - validating, 6-4
  - value, 5-14
- configuration, 5-4
  - accessing menu, 5-2
  - adding a recipe, 5-5
  - copying a recipe, 5-6
  - deleting a recipe, 5-6
  - methods, 2-2
  - modifying a recipe, 5-8
  - on two stations, 1-7
  - procedures, 1-8

Recipe
- configuration (continued)
  - recipe area, 1-10, 3-2, 3-4
  - recipe area tag, 1-9
- control block, 4-4
  - nesting, 4-6
  - phased download, 4-10
  - RCB_BEG, 4-5
  - RCB_END, 4-5
  - RCB_WAIT, 4-4
- creating, 1-13
- directory, 7-2
- download
  - by controller, 7-14
    - canceling, 7-14
    - modifying, 7-15
    - selecting, 7-14
    - starting, 7-14
  - manually, 7-11
    - canceling, 7-13
    - continuing, 7-13
    - selecting, 7-2
    - starting, 7-13
    - suspending, 7-13
- error messages, example, 6-5
- handshake, example, 4-8
- linking recipe area, 5-15
- linking to recipe area, 1-14
- loading, 7-6
  - troubleshooting, 7-10
- operator change log, 7-11
- overview, 1-2
- phased download, 4-10
- printing, 7-13
- production value, modifying, 7-11
- scale factor, 5-13
  - modifying, 7-12
  - troubleshooting, 7-8
- subrecipe, 1-6, 4-2
- validating, 1-15, 6-2
  - error messages, 6-6
- value, scaling, 7-12

Recipe area, linking recipe, 5-15

S

Scale factor, recipe components, 7-12
  troubleshooting, 7-8

Subrecipe, 1-6, 4-2

T

Tag, types, area (AREA), 1-9

Troubleshooting
  loading recipes, 7-8, 7-10
  recipe download, 7-13
  recipe load, 7-10
  recipe validation errors, 6-5
  scale factor, 7-8

V

Validating, recipes, 6-4

Customer Response

We would like to know what you think about our user manuals so that we can serve you better. How would you rate the quality of our manuals?

	Excellent	Good	Fair	Poor
Accuracy	_____	_____	_____	_____
Organization	_____	_____	_____	_____
Clarity	_____	_____	_____	_____
Completeness	_____	_____	_____	_____
Graphics	_____	_____	_____	_____
Examples	_____	_____	_____	_____
Overall design	_____	_____	_____	_____
Size	_____	_____	_____	_____
Index	_____	_____	_____	_____

Would you be interested in giving us more detailed comments about our manuals?

☐ **Yes!** Please send me a questionnaire.

☐ **No.** Thanks anyway.

Your Name: _____

Title: _____

Telephone Number: (____) _____

Company Name: _____

Company Address: _____

_____

_____

Manual Name: SIMATIC PCS 7 OSx 4.1.2 Recipe Manual **Edition:** Original
Manual Assembly Number: 2811150-0001 **Date:** 7/02
Order Number: 6ES7 6550XX058BC3

FOLD

SIEMENS ENERGY & AUTOMATION INC
3000 BILL GARLAND ROAD
P O BOX 1255
JOHNSON CITY TN 37605-1255

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO.3 JOHNSON CITY, TN

POSTAGE WILL BE PAID BY ADDRESSEE

ATTN TECHNICAL COMMUNICATIONS M/S 519
SIEMENS ENERGY & AUTOMATION INC
P O BOX 1255
JOHNSON CITY TN 37605-1255

FOLD